

Complexity

Advances in Processing, Mining, and Learning Complex Data: From Foundations to Real-World Applications

Lead Guest Editor: Jia Wu

Guest Editors: Shirui Pan, Chuan Zhou, Gang Li, Wu He, and Chengqi Zhang





**Advances in Processing, Mining, and Learning
Complex Data: From Foundations to Real-World
Applications**

Complexity

Advances in Processing, Mining, and Learning Complex Data: From Foundations to Real-World Applications

Lead Guest Editor: Jia Wu

Guest Editors: Shirui Pan, Chuan Zhou, Gang Li, Wu He,
and Chengqi Zhang



Copyright © 2018 Hindawi. All rights reserved.

This is a special issue published in “Complexity.” All articles are open access articles distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Editorial Board

- José A. Acosta, Spain
Carlos F. Aguilar-Ibáñez, Mexico
Tarek Ahmed-Ali, France
Basil M. Al-Hadithi, Spain
Juan A. Almendral, Spain
Diego R. Amancio, Brazil
David Arroyo, Spain
Mohamed Boutayeb, France
Arturo Buscarino, Italy
Guido Caldarelli, Italy
Eric Campos-Canton, Mexico
Mohammed Chadli, France
Diyi Chen, China
Giulio Cimini, Italy
Danilo Comminiello, Italy
Sara Dadras, USA
Manlio De Domenico, Italy
Pietro De Lellis, Italy
Albert Diaz-Guilera, Spain
Thach Ngoc Dinh, France
Jordi Duch, Spain
Marcio Eisencraft, Brazil
Joshua Epstein, USA
Mondher Farza, France
Thierry Floquet, France
Mattia Frasca, Italy
Lucia Valentina Gambuzza, Italy
Bernhard C. Geiger, Austria
- Carlos Gershenson, Mexico
Peter Giesl, UK
Sergio Gómez, Spain
Wesley N. Gonçalves, Brazil
Lingzhong Guo, UK
Xianggui Guo, China
Sigurdur F. Hafstein, Iceland
Chittaranjan Hens, Israel
Giacomo Innocenti, Italy
Sarangapani Jagannathan, USA
Mahdi Jalili, Australia
Jeffrey H. Johnson, UK
M. Hassan Khooban, Denmark
Vincent Labatut, France
Lucas Lacasa, UK
Qingdu Li, Germany
Chongyang Liu, China
Xiaoping Liu, Canada
Rosa M. Lopez Gutierrez, Mexico
Vittorio Loreto, Italy
Didier Maquin, France
Eulalia Martínez, Spain
Marcelo Messias, Brazil
Ana Meštrović, Croatia
Ludovico Minati, Japan
Ch. P. Monterola, Philippines
Marcin Mrugalski, Poland
Roberto Natella, Italy
- Nam-Phong Nguyen, USA
Beatrice M. Ombuki-Berman, Canada
Irene Otero-Muras, Spain
Yongping Pan, Singapore
Daniela Paolotti, Italy
Mahardhika Pratama, Singapore
Luis M. Rocha, USA
Miguel Romance, Spain
Avimanyu Sahoo, USA
Matilde Santos, Spain
Hiroki Sayama, USA
Michele Scarpiniti, Italy
Enzo Pasquale Scilingo, Italy
Dan Selișteanu, Romania
Dimitrios Stamovlasis, Greece
Samuel Stanton, USA
Roberto Tonelli, Italy
Shahadat Uddin, Australia
Gaetano Valenza, Italy
Dimitri Volchenkov, USA
Christos Volos, Greece
Zidong Wang, UK
Yan-Ling Wei, Singapore
Honglei Xu, Australia
Xinggang Yan, UK
Massimiliano Zanin, Spain
Hassan Zargarzadeh, USA
Rongqing Zhang, USA

Contents

Advances in Processing, Mining, and Learning Complex Data: From Foundations to Real-World Applications

Jia Wu , Shirui Pan , Chuan Zhou, Gang Li, Wu He, and Chengqi Zhang
Editorial (3 pages), Article ID 7861860, Volume 2018 (2018)

Background Information Self-Learning Based Hyperspectral Target Detection

Yufei Tian, Jihai Yang, Shijun Li , and Wenning Xu 
Research Article (7 pages), Article ID 3502508, Volume 2018 (2018)

Unsupervised Domain Adaptation Using Exemplar-SVMs with Adaptation Regularization

Yiwei He , Yingjie Tian , Jingjing Tang , and Yue Ma 
Research Article (13 pages), Article ID 8425821, Volume 2018 (2018)

Exploiting Interslice Correlation for MRI Prostate Image Segmentation, from Recursive Neural Networks Aspect

Qikui Zhu, Bo Du , Baris Turkbey, Peter Choyke, and Pingkun Yan
Research Article (10 pages), Article ID 4185279, Volume 2018 (2018)

Transferable Feature Representation for Visible-to-Infrared Cross-Dataset Human Action Recognition

Yang Liu , Zhaoyang Lu, Jing Li, Chao Yao , and Yanzi Deng
Research Article (20 pages), Article ID 5345241, Volume 2018 (2018)

Efficient Privacy-Preserving Protocol for k -NN Search over Encrypted Data in Location-Based Service

Huijuan Lian, Weidong Qiu, Di Yan, Zheng Huang, and Jie Guo
Research Article (14 pages), Article ID 1490283, Volume 2017 (2018)

Privacy-Preserving and Scalable Service Recommendation Based on SimHash in a Distributed Cloud Environment

Yanwei Xu, Lianyong Qi, Wanchun Dou, and Jiguo Yu
Research Article (9 pages), Article ID 3437854, Volume 2017 (2018)

Kernel Negative ϵ Dragging Linear Regression for Pattern Classification

Yali Peng, Lu Zhang, Shigang Liu, Xili Wang, and Min Guo
Research Article (14 pages), Article ID 2691474, Volume 2017 (2018)

Mining Community-Level Influence in Microblogging Network: A Case Study on Sina Weibo

Yufei Liu, Dechang Pi, and Lin Cui
Research Article (16 pages), Article ID 4783159, Volume 2017 (2018)

Deep Recurrent Model for Server Load and Performance Prediction in Data Center

Zheng Huang, Jiajun Peng, Huijuan Lian, Jie Guo, and Weidong Qiu
Research Article (10 pages), Article ID 8584252, Volume 2017 (2018)

Deep Hierarchical Representation from Classifying Logo-405

Sujuan Hou, Jianwei Lin, Shangbo Zhou, Maoling Qin, Weikuan Jia, and Yuanjie Zheng
Research Article (12 pages), Article ID 3169149, Volume 2017 (2018)



A Distributed K -Means Segmentation Algorithm Applied to *Lobesia botrana* Recognition

José García, Christopher Pope, and Francisco Altimiras

Research Article (14 pages), Article ID 5137317, Volume 2017 (2018)

Determining the Coupling Source on a Set of Oscillators from Experimental Data

Juan Carlos Jauregui-Correa, Carlos S. López-Cajun, and Mihir Sen

Research Article (10 pages), Article ID 8017138, Volume 2017 (2018)

Editorial

Advances in Processing, Mining, and Learning Complex Data: From Foundations to Real-World Applications

Jia Wu ¹, Shirui Pan ², Chuan Zhou,³ Gang Li,⁴ Wu He,⁵ and Chengqi Zhang²

¹Macquarie University, Sydney, NSW, Australia

²University of Technology Sydney, Sydney, NSW, Australia

³Chinese Academy of Sciences, Beijing, China

⁴Deakin University, Melbourne, VIC, Australia

⁵Old Dominion University, Norfolk, VA, USA

Correspondence should be addressed to Jia Wu; jia.wu@mq.edu.au

Received 17 May 2018; Accepted 17 May 2018; Published 19 July 2018

Copyright © 2018 Jia Wu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Processing, mining, and learning complex data refer to an advanced study area of data mining and knowledge discovery concerning the development and analysis of approaches for discovering patterns and learning models from data with a complex structure (e.g., multirelational data, XML data, text data, image data, time series, sequences, graphs, streaming data, and trees) [1–5]. These kinds of data are commonly encountered in many social, economic, scientific, and engineering applications. Complex data pose new challenges for current research in data mining and knowledge discovery as they require new methods for processing, mining, and learning them. Traditional data analysis methods often require the data to be represented as vectors [6]. However, many data objects in real-world applications, such as chemical compounds in biopharmacy, brain regions in brain health data, users in business networks, and time-series information in medical data, contain rich structure information (e.g., relationships between data and temporal structures). Such a simple feature-vector representation inherently loses the structure information of the objects. In reality, objects may have complicated characteristics, depending on how the objects are assessed and characterized. Meanwhile, the data may come from heterogeneous domains [7], such as traditional tabular-based data, sequential patterns, graphs, time-series information, and semistructured data. Novel data analytics methods are desired to discover meaningful

knowledge in advanced applications from data objects with complex characteristics. This special issue contributes to the fundamental research in processing, mining, and learning complex data, focusing on the analysis of complex data sources.

1. Spatial Data

With the development of mobile communication technology, location-based services are booming prosperously. Meanwhile, privacy protection has become the main obstacle for the further development of services. The paper titled “Efficient Privacy-Preserving Protocol for k -NN Search over Encrypted Data in Location-Based Service” proposes an efficient private circular query protocol with high accuracy rate and low computation and communication cost. The Moore curve is adopted to convert two-dimensional spatial data into one-dimensional sequence and encrypt the points of interest (POIs) information with the Brakerski-Gentry-Vaikuntanathan homomorphic encryption scheme for privacy preserving. The scheme performs the secret circular shifts of the encrypted POI information to hide the location of the user without a trusted third party. The proposed scheme provides high-accuracy query results while maintaining low computation and communication cost.

2. Web Server Data

The paper titled “Deep Recurrent Model for Server Load and Performance Prediction in Data Center” proposes to use deep learning to predict web server performance and workload. The model can extract features automatically during the learning process without any prior knowledge or hand-generated features for segmentation. Experiments conducted on real web server data sets show that the model can achieve a good performance and generalization on predicting the performance of different kinds of servers. And the result also shows that the load generated by our model is very similar to the real one, which can be applied to test data center and other kinds of servers. Most servers in data center have log system. As long as the log file recording of the operation of the users is provided, the method can be used to generate load for the server and predict server performance under different load conditions.

3. Image Data

The paper titled “Unsupervised Domain Adaptation Using Exemplar-SVMs with Adaptation Regularization” has proposed an effective method for domain adaptation problems with regularization item which reduces the data distribution mismatch between domains and preserves properties of the original data. Furthermore, utilizing the method of integrating classifiers can predict target domain data with high accuracy. The proposed method mainly aims to predict in the setting that exists distribution mismatch across domains or instances and achieves desired results. Experiments conducted on the transfer learning datasets transfer knowledge from image to image.

Hyperspectral imaging has been proved as an effective way to explore the useful information behind the land objects. And it can also be adopted for biologic information extraction, by which the origin information can be acquired from the image repeatedly without contamination. The paper titled “Background Information Self-Learning Based Hyperspectral Target Detection” proposes a target detection method based on background self-learning to extract the biologic information from the hyperspectral images. The conventional unstructured target detectors are very difficult to estimate the background statistics accurately neither in a global nor local way. Considering the spatial spectral information, its performance can be further improved by avoiding the above problem. It is especially designed to extract fingerprint and tumor region from hyperspectral biologic images. The validity and the superiority of the method have been demonstrated on detecting the biologic information from hyperspectral images.

Segmentation of the prostate from magnetic resonance imaging plays an important role in prostate cancer diagnosis. However, the lack of clear boundary and significant variation of prostate shapes and appearances make the automatic segmentation very challenging. In the past several years, approaches based on deep learning technology have made significant progress on prostate segmentation. However, those approaches mainly paid attention to features and

contexts within each single slice of a 3D volume. As a result, these kinds of approaches face many difficulties when segmenting the base and apex of the prostate due to the limited slice boundary information. To tackle this problem, the paper titled “Exploiting Inter-Slice Correlation for MRI Prostate Image Segmentation: From Recursive Neural Networks Aspect” proposes a deep neural network with bidirectional convolutional recurrent layers for magnetic resonance imaging of prostate image segmentation. In addition to utilizing the intraslice contexts and features, the proposed model also treats prostate slices as a data sequence and utilizes the interslice contexts to assist segmentation. The proposed approach achieved significant segmentation improvement compared to other reported methods.

Early detection of *Lobesia botrana* is a primary issue for a proper control of this insect considered as the major pest in grapevine. The paper titled “A Distributed K -Means Segmentation Algorithm Applied to *Lobesia botrana* Recognition” proposes a novel method for *L. botrana* recognition using image data mining based on clustering segmentation with descriptors which consider gray scale values and gradient in each segment. This system allows a 95 percent of *L. botrana* recognition in nonfully controlled lighting, zoom, and orientation environments. The image capture application is currently implemented in a mobile application, and subsequent segmentation processing is done in the cloud.

The paper titled “Deep Hierarchical Representation from Classifying Logo405” introduces a logo classification mechanism which combines a series of deep representations obtained by fine-tuning convolutional neural network architectures and traditional pattern recognition algorithms. The experiments are carried out on both the Logo-405 dataset and the publicly available FlickrLogos-32 image datasets. The experimental results demonstrate that the proposed mechanism outperforms two popular ways used for logo classification, including the strategies that integrate hand-crafted features and traditional pattern recognition algorithms and the models.

The paper titled “Kernel Negative ϵ Dragging Linear Regression for Pattern Classification” proposes a kernel negative ϵ dragging linear regression method for pattern classification, which simultaneously integrated the negative ϵ dragging technique and the kernel method into linear regression for robust pattern classification under the condition that the consistency and compatibility between the test samples and training samples are poor. The negative ϵ dragging technique learns a classifier with a proper margin from noised and deformable data. Meanwhile, the kernel approach can make linearly nonseparable samples become linearly separable. Based on the effect of the negative ϵ dragging technique and kernel collaborating, the method can better perform classification for noised and deformable data. Comprehensive 24 experiments on image data sets demonstrate algorithm performance.

Recently, infrared human action recognition has attracted increasing attention for it has many advantages over visible light, that is, robust to illumination change and shadows. However, the infrared action data is limited until

now, which degrades the performance of infrared action recognition. Motivated by the idea of transfer learning, an infrared human action recognition framework using auxiliary data from visible light is proposed to solve the problem of limited infrared action data in the paper titled “Transferable Feature Representation for Visible-to-Infrared Cross-Dataset Human Action Recognition.” The proposed method is evaluated on InfAR, which is a publicly available infrared human action dataset. To build up auxiliary data, we set up a novel visible light action dataset XD145. Experimental results show that the proposed method can achieve state-of-the-art performance compared with several transfer learning and domain adaptation methods.

4. Social Network Data

Social influence analysis is important for many social network applications, including recommendation and cyber security analysis. The influence of community including multiple users outweighs the individual influence. Existing models focus on the individual influence analysis, but few studies estimate the community influence that is ubiquitous in online social network. A major challenge lies in that researchers need to take into account many factors, such as user influence, social trust, and user relationship, to model community-level influence. The paper titled “Mining Community-Level Influence in Microblogging Network: A Case Study on Sina Weibo” aims to assess the community-level influence effectively and accurately; the problem is formulated as modeling community influence and constructs a community-level influence analysis model. Empirical studies on a real-world dataset from Sina Weibo demonstrate the superiority of the proposed model.

In the distributed cloud environment, a cloud platform is often not willing to share its recorded user-service invocation data with other cloud platforms due to privacy concerns, which decreases the feasibility of cross-cloud collaborative service recommendation severely. Besides, the user-service invocation data recorded by each cloud platform may update over time, which reduces the recommendation scalability significantly. In view of these two challenges, a novel privacy-preserving and scalable service recommendation approach based on SimHash, that is, SerRecSimHash, is put forward in the paper titled “Privacy-Preserving and Scalable Service Recommendation Based on SimHash in A Distributed Cloud Environment.” A set of experiments are conducted based on a real distributed service quality dataset WS-DREAM. Experiment results show that SerRecSimHash outperforms the other up-to-date approaches in terms of recommendation accuracy and efficiency while guaranteeing privacy preservation.

5. Time Series/Signal Data

Complex systems is a broad concept that comprises many disciplines, including engineering systems. Regardless of their particular behavior, complex systems share similar behaviors, such as synchronization. The paper titled “Determining the Coupling Source on a Set of Oscillators from

Experimental Data” presents different techniques for determining the source of coupling when a set of oscillators synchronize. It is possible to identify the location and time variations of the coupling by applying a combination of analysis techniques, namely, the source of synchronization. For this purpose, the analysis of experimental data from a complex mechanical system is presented. The experiment consisted of placing a 24-bladed rotor under an airflow. The vibratory motion of the blades was recorded with accelerometers, and the resulting information was analyzed with four techniques: correlation coefficients, Kuramoto parameter, cross-correlation functions, and the recurrence plot. The measurements clearly show the existence of frequencies due to the foreground components and the internal interaction between them due to the background components (coupling).

Acknowledgments

The Guest Editorial Team would like to express their gratitude to all the authors for their interest in selecting this special issue as a venue for their scholarly work dissemination. The editors also wish to thank the anonymous reviewers for their careful reading of the manuscripts submitted to this special issue collection and their many insightful comments and suggestions.

Jia Wu
Shirui Pan
Chuan Zhou
Gang Li
Wu He
Chengqi Zhang

References

- [1] M. Shi, J. Liu, D. Zhou, and Y. Tang, “A topic-sensitive method for mashup tag recommendation utilizing multi-relational service data,” *IEEE Transactions on Services Computing*, p. 1, 2018.
- [2] H. Fan, Z. Ma, D. Wang, and J. Liu, “Handling distributed XML queries over large XML data based on MapReduce framework,” *Information Sciences*, vol. 453, pp. 1–20, 2018.
- [3] C. X. Zhai, “Probabilistic topic models for text data retrieval and analysis,” in *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval - SIGIR '17*, pp. 1399–1401, Tokyo, Japan, 2017.
- [4] L. B. Godfrey and M. S. Gashler, “Neural decomposition of time-series data for effective generalization,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, no. 7, pp. 1–13, 2017.
- [5] H. Meyerhenke, P. Sanders, and C. Schulz, “Parallel graph partitioning for complex networks,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 28, no. 9, pp. 2625–2638, 2017.
- [6] K. Thearling, *An Introduction to Data Mining*, 2017.
- [7] Y. Luo, Y. Wen, T. Liu, and D. Tao, “Transferring knowledge fragments for learning distance metric from a heterogeneous domain,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, p. 1, 2018.

Research Article

Background Information Self-Learning Based Hyperspectral Target Detection

Yufei Tian,^{1,2} Jihai Yang,² Shijun Li ,² and Wenning Xu ³

¹Hubei Bosheng Digital Education Service Co., Ltd., Wuhan, China

²School of Computer, Wuhan University, Wuhan, China

³Chinese Academy of Geological Sciences, Beijing, China

Correspondence should be addressed to Wenning Xu; xwncags@163.com

Received 24 August 2017; Revised 30 January 2018; Accepted 28 February 2018; Published 3 July 2018

Academic Editor: Jia Wu

Copyright © 2018 Yufei Tian et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Hyperspectral imaging has been proved as an effective way to explore the useful information behind the land objects. And it can also be adopted for biologic information extraction, by which the origin information can be acquired from the image repeatedly without contamination. In this paper we proposed a target detection method based on background self-learning to extract the biologic information from the hyperspectral images. The conventional unstructured target detectors are very difficult to estimate the background statistics accurately in either a global or local way. Considering the spatial spectral information, its performance can be further improved by avoiding the above problem. It is especially designed to extract fingerprint and tumor region from hyperspectral biologic images. The experimental results show the validity and the superiority of our method on detecting the biologic information from hyperspectral images.

1. Introduction

Physical direct extraction method is commonly used as a traditional method to extract the biologic information attached or to make sure that there is any abnormal characteristic information existing in body surface or a body region, and it will always cause damage to the information carrier. Recent works on graph theory and its applications draw great attentions [1–3]. There is a reliable method [4] to obtain the information which is capturing an image with the target characteristic information and processing the image. In general, the image-based biologic feature extraction method is mostly a single-band full-color image, which obtains target region shape [5–9] by calculating the shape of the image or filtering enhancement of the image. However, there are obvious defects in the traditional single-band full-color image. The characteristics of single-band full-color image will change greatly in the case of existing differences in the external environment such as the angle of light and other factors, which increases the difficulty of extraction. With the advent of imaging spectrometers, we can obtain hyperspectral images with many bands, continuous spectral

curves, and consistent features [10]. We can obtain more abundant and stable information by using hyperspectral technology to image for biological carrier, and it facilitates the further extraction of biological information. As for the extraction of some characteristic information on the biological carrier, it can be done by target detection [11]. In recent years, many object extraction and target detection methods for hyperspectral images have emerged. The most common methods are matched filter (MF) method [12], Constrained Energy Minimization (CEM) method [13], Adaptive Cosine Consistency Estimator (ACE) method [14], Spectral Angle Method (SAM) [15], Orthogonal Subspace Projection (OSP) method [16], and so on. Among these methods, the MF and SAM methods do not suppress the background information. The CEM, ACE, and OSP methods suppress the background information while being affected by the target information, resulting in that the estimation of the background information is not accurate enough. In this paper, we propose a Background Self-Learning Target Detection (BSLD) algorithm for hyperspectral image, which makes the target information to be extracted more prominent by suppressing the background information accurately.

2. Information Extraction Methods by Background Information Self-Learning

Target information extraction is a process of separating the target area from the nontarget area, that is, the background information. In order to get the difference between the target information and the background information, the key is to calculate the accurate background estimation information, which directly determines the performance of the information extraction algorithm. The method, such as ACE, based on unstructured background is a common method of hyperspectral image object extraction. It is assumed that the not interested region is homogeneous and can be represented by a multivariate normal distribution, and, by using the sample data, we could estimate the background covariance matrix if the background information and a priori target information are known [17]. Kelly proposed the generalized likelihood ratio structure detection algorithm which is based on unstructured background firstly, and we obtain adaptive cosine consistency evaluation ACE and adaptive matched filter AMF on this basis [18]. These methods are calculated for the background covariance estimates directly from the entire image data, but this will cause some error due to the absence of the object information to be extracted. In recent years, many local background estimation methods have emerged, which take into account the spatial information in the data to improve the accuracy of background information estimation. One of the common methods is to segment the image and then select the space closest to the test pixel to estimate the background of the statistical data [19]. Another common way is to use a sliding window, by limiting the size of the window area, and, at the same time, also need to introduce the “internal window” to exclude information that may not be background. In this way, not only will the size of the inside window and outside window affect the effect of the whole algorithm and the calculation of different sizes, but also irregular area background statistics is very time-consuming, affecting the implementation efficiency of the algorithm.

To solve the above problems, according to different background spectral information to construct different multiple normal distribution model, this paper will propose a background self-learning method. The background pixel is clustered according to the spectral information, and the background type structure detector is determined based on the spectral information of the test pixel and the pixel point associated with its spatial when the information is extracted. The method based on the background self-learning framework can use all kinds of statistical information flexibly and suppress the background better and highlight the target to be extracted. At the same time, it will reduce the computational complexity of the above method. Specifically, the self-learning of the background information presented in this paper is divided into the following five steps.

2.1. Estimating the Number of End Elements in the Image. The end element is also called the basic component unit. The spectral information of each end element can approximately represent a class of signals in the image. The minimum error hyperspectral signal identification method (HySime method)

was used to estimate the number of end elements in the image [20]. The HySime method inputs the original hyperspectral image named d and its autocorrelation matrix:

$$R_d = \frac{1}{n} (d^T d). \quad (1)$$

The noise is estimated by noise and the noise autocorrelation matrix is calculated:

$$R_\xi = \frac{1}{n} \sum_{i=1}^n (\xi_i^T \xi_i). \quad (2)$$

And then calculate the autocorrelation matrix of original image which removed the noise information:

$$R_x = \frac{1}{n} \sum_{i=1}^n [(d_i - \xi_i)^T (d_i - \xi_i)]. \quad (3)$$

By the above formula to get the characteristics of the group named R_x , $E \equiv [e_1, \dots, e_k]$, we could use the method to propose the error minimization formula:

$$\begin{aligned} (k, \pi) &= \arg \min_{k, \pi} \{ \text{tr}(U_k^\perp R_d) + 2 \text{tr}(U_k R_\xi) \} \\ &= \arg \min_{k, \pi} \left\{ c + \sum_{j=1}^k -p_{ij} + 2\sigma_{ij}^2 \right\}. \end{aligned} \quad (4)$$

Calculate the subspace dimension of the image, that is, the estimation of the number k of end elements, where the k dimensions space is specified:

$$\begin{aligned} E_k &\equiv [e_{i1}, \dots, e_{ik}] \\ U_k &= E_k^T E_k \\ U_k^\perp &= I - U_k. \end{aligned} \quad (5)$$

c is not related to constant; p_{ij} and σ_{ij}^2 are expressed as

$$\begin{aligned} p_{ij} &= e_{ij}^T R_d e_{ij} \\ \sigma_{ij}^2 &= e_{ij}^T R_\xi e_{ij}. \end{aligned} \quad (6)$$

The resulting end number k is the subsequent parameter of the K -means method in Step 3.2.

2.2. Background Clustering Fusion. We have to cluster the image in order to get the statistical information of the background subspace formed by the same background element. The number of end elements has been obtained by HySime method in Step 3.1, and the background information is clustered by introducing K -means method as cluster number. Assume that we have n data points which need to be clustered into k classes, and, in this paper, K -means is not described in detail:

$$J = \sum_{i=1}^n \sum_{j=1}^k (r_{ij} \|data_i - \mu_j\|^2). \quad (7)$$

The value of r_{ij} is 1 when $data_i$ are clustered; otherwise, it is 0. In this paper, K -means detailed solution is not too much; as for the specific implementation method, they could be seen in [21].

After obtaining the clustering information, the spectral information of the cluster center is regarded as the vector, the spectral angle θ_{ij} among clusters is calculated, and the spectral angle merge is calculated if $\theta_{ij} < T_1$ is true. For the center of two clusters which are c_i, c_j , the spectral angle can be calculated by the following formula:

$$\theta_{ij} = \arccos \frac{c_i \cdot c_j^T}{\|c_i\| \times \|c_j\|}. \quad (8)$$

In order to avoid generating morbid matrices during the process of calculating the covariance matrix, we merge the classes with few pixels into the nearest class at the same time. In this experiment, the number of images' bands has to be no more than the number of each cluster's pixels. After the above adjustment, we can get the clustering information we need and calculate the covariance matrix of each cluster on this basis.

2.3. Excluding Target Classes. To avoid the nonbackground information (target information) being clustered into the background class collection, we find the closest class to the previous information spectral feature and compare the cluster center with the spectral angle θ_2 of the previous information by formula (8). If $\theta_2 < T_2$, then it is determined as the target information class, and we need to abandon the class; otherwise, there is no similar class to the prior information. Skip this step.

2.4. The Choice of the Best Background Class. Before extracting the target information, we should determine a background class for each pixel firstly. And the statistics of the background class will be used in the process of the subsequent target information extraction calculation. There are two ways to determine the background class for each pixel, as is shown in Figure 1. One is a method of selecting a background class based on a previous information, which selects the category of the cluster center that is closest to the prior spectral information as background estimation. When we determine the appropriate background class for the test pixel, if the dark red clustering center is closest to the orange prior information, then the background class is chosen to calculate the covariance matrix, so that it can suppress alarm pixels better which are too similar to previous information to distinguish with it.

The other method is to select the background class based on the test pixels, which chooses the class closest to clustering center of the test pixel spectral information as background estimation. In Figure 1, the green test pixel is closest to the blue clustering center, so we decided to use the blue background class to calculate the background information of the test pixel. For different locations of the pixel, with different background statistics, the latter is more feasible than the former. Through the above steps, we can get the background pixel set and obtain the background statistical covariance Σ .

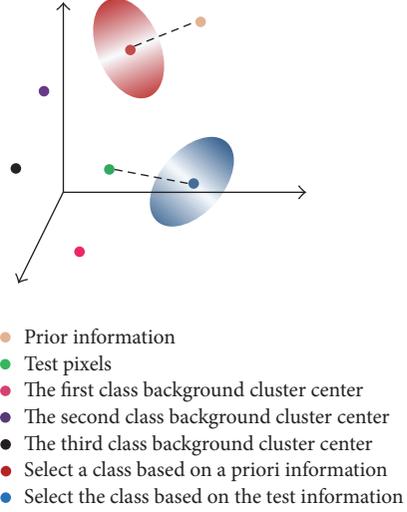


FIGURE 1: Illustration determined background class.

$\frac{1}{\sqrt{8}}$	$\frac{1}{\sqrt{5}}$	$\frac{1}{2}$	$\frac{1}{\sqrt{5}}$	$\frac{1}{\sqrt{8}}$
$\frac{1}{\sqrt{5}}$	$\frac{1}{\sqrt{2}}$	1	$\frac{1}{\sqrt{2}}$	$\frac{1}{\sqrt{5}}$
$\frac{1}{2}$	1	w	1	$\frac{1}{2}$
$\frac{1}{\sqrt{5}}$	$\frac{1}{\sqrt{2}}$	1	$\frac{1}{\sqrt{2}}$	$\frac{1}{\sqrt{5}}$
$\frac{1}{\sqrt{8}}$	$\frac{1}{\sqrt{5}}$	$\frac{1}{2}$	$\frac{1}{\sqrt{5}}$	$\frac{1}{\sqrt{8}}$

FIGURE 2: Sliding window weight, size = 2.

2.5. Target Information Extraction. In order to describe the pixel information more accurately, we also need make use of the spatial information of the pixel spectrum in the image. Here, we introduce the concept of sliding window, that is, selecting the current pixel as the center, delineating a square area, and all the pixels' information in the region determines the current pixel background class together, just as is shown in Figure 2. We should use the reasonable spatial information to avoid the spectral fluctuations caused by the pixel and unknown information and abnormal mixed information, so we could have a more realistic description of the real pixel spectral information test. Conventional calculation method is time-consuming, so this paper takes the method that averages all the pixels on the window, the window size can be adjusted according to the actual situation, the value of $size$ often uses 1, 2, 3, and so on, and each pixel of the window has an assigned value:

$$w_{ij} = \frac{1}{\sqrt{(x-i)^2 + (y-j)^2}}. \quad (9)$$

(x, y) indicates the center location of the window, $i \in [x - size, x + size]$, $i \neq x$; $j \in [y - size, y + size]$, $j \neq y$.

Input threshold T_1, T_2 and obtain the size of window and the weight of window center element, using the method proposed in this paper to complete the target information extraction process shown in Figure 3.

Weighted average of pixels within the window is as follows:

$$w_{d_{ij}} = \frac{\sum_{i=x-size}^{x+size} \sum_{j=y-size}^{y+size} w_{ij} d_{ij}}{\sum_{i=x-size}^{x+size} \sum_{j=y-size}^{y+size} w_{ij}}. \quad (10)$$

After obtaining the spectral information of the test pixel with spatial information, the generalized likelihood ratio method can be used to express the algorithm:

$$D_{BSL}(x) = \frac{(t^T \Sigma^{-1} x)^2}{(t^T \Sigma^{-1} t)(x^T \Sigma^{-1} x)}. \quad (11)$$

x is the spectral pixel of the test pixel. The weighted average result x in (10) is taken into (11), which is the background spectral information, which is the background class covariance Σ determined by the test pixels. Each pixel is calculated from the above formula, and the resulting value is determined by threshold division as the target information we need to extract.

Assume that the original hyperspectral image data is *data*.

3. Experiment and Analysis

To verify the feasibility and effectiveness of the method described in this paper, two sets of hyperspectral biological images were used in this paper. We compare the BSLD method with the adaptive cosine coincidence (ACE), the constrained energy minimization method (CEM), and the matched filter.

3.1. Indicators of Evaluation Performance. In this paper, ROC (Receiver Operating Characteristic) curve is used to evaluate the performance of the algorithm. The ROC curve is a curve that calculates the correct extraction rate and the false alarm rate corresponding to the point of the unit coordinate system. In Section 2.5, we obtain a decision value for each pixel by the extraction decision algorithm. Through changing thresholds, the correct extraction rate and false alarm rate of the algorithm are obtained. The higher the threshold, the lower the correct extraction rate and the false alarm rate. We hope to obtain a higher correct extraction rate in the case of low false alarm rate. In theory, that is, the more curve to the upper left corner of the protrusion, the better the performance of the algorithm. Another comparison method is to compare the area under the curve (AUC) (area under the curve of ROC); when the value of AUC is larger, the performance of the algorithm is better [22].

3.2. Experiment of Fingerprint Information Extraction. The fingerprint information extraction experiment data is an envelope local hyperspectral image with a band number of 31 and a size of $640 * 512$. As is shown in Figure 4, the gray-scale

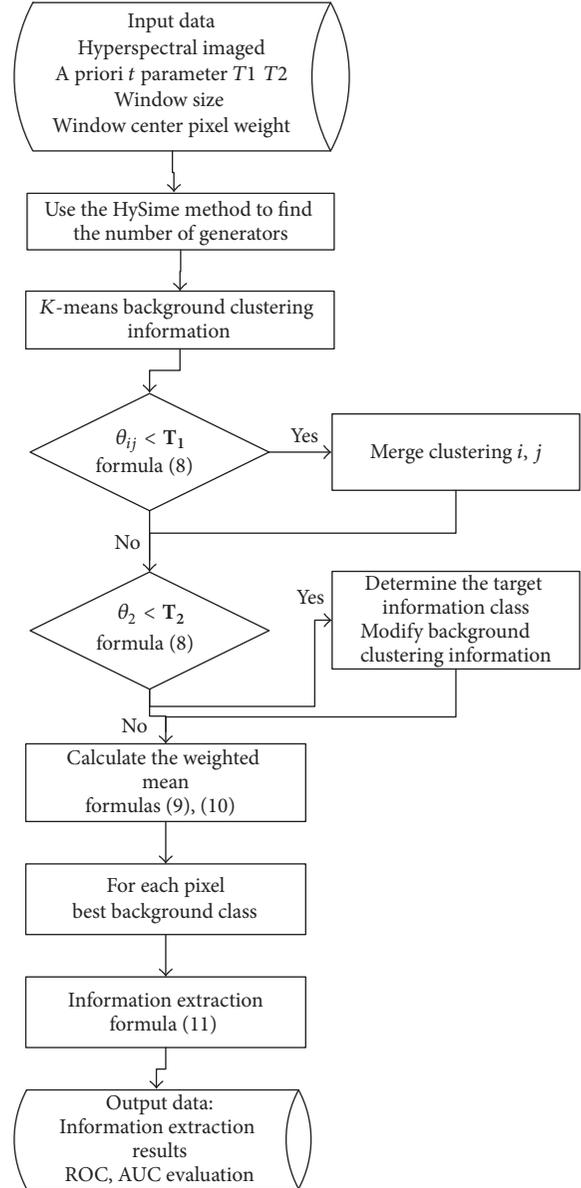


FIGURE 3: BSLD method flow chart.

images of the bands are 1, 11, 21, and 31, respectively. From the gray-scale point of view, only the first band can clearly identify the fingerprint, but it is also overlapped with the background information or block, so it greatly enhances the difficulty of fingerprint shape extraction.

According to the data provided by a total of 8694-pixel ROI (Regions of Interest) [23] as the correct fingerprint extracting reference value, we selected pixel (*Sample, Line*) = (290, 147) as previous information to experiment. Taking the experimental parameters, the window size is 5. The best experimental results can be obtained when the window center pixel weight is 3. As is shown in Figure 5, the fingerprint information can be clearly extracted except the area covered by the ink, and the area covered by the ink can also be extracted partly. In Figure 6, we can see that the BSLD method (red curve) is at the top of the ROC curve of other methods.

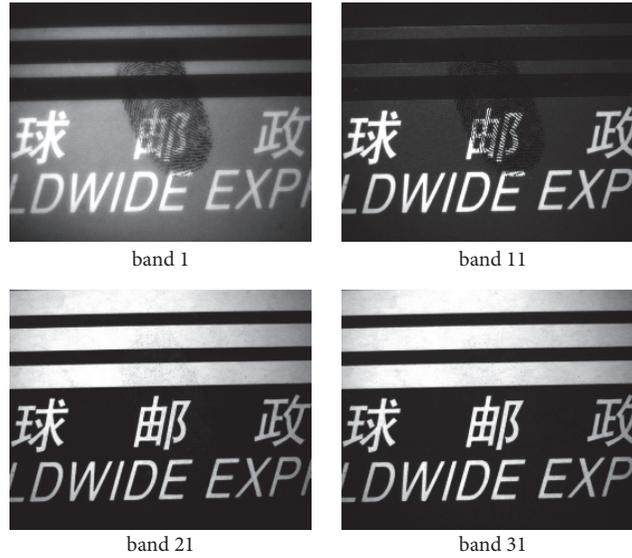


FIGURE 4: Four bands in the fingerprint data.

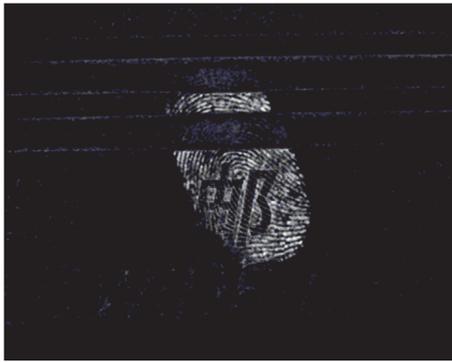


FIGURE 5: Fingerprint information extraction effect.

The AUC value in Table 1 also reflects that this method has the best fingerprint information extraction performance.

3.3. Experimental Study on Tumor Area Information Extraction in Medical Images. In the tumor area information extraction experiment, the data we selected is a hyperspectral image of the trunk part of the mouse in which band's number is 13, spectral resolution is 5 nm, and the size is 220 * 200. In Figure 7, the gray-scale images of bands 1 and 5 and the color images of the red, green, and blue bands are shown. There is silver powder on the back and accumulated food in the abdomen. This data gives the samples of prostate tumor cells, but it cannot clearly identify the exact and complete position of the tumor area from the band image. The region is extracted exactly and accurately. The region is extracted accurately through this algorithm.

As is shown in Figure 8, it is not difficult to find that the distribution information of the prostate area from the hind leg root upward is very obvious, and the tumor area which is difficult to be positioned in the original hyperspectral image is clearly extracted. Although there are some false alarms

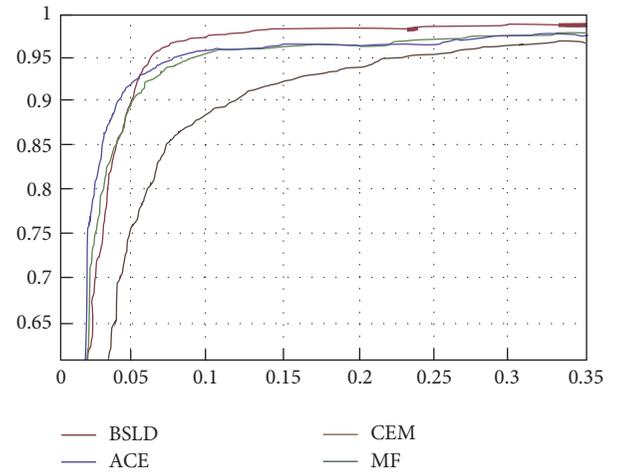


FIGURE 6: ROC curve evaluation of fingerprint information extraction.

TABLE 1: Evaluation of AUC of fingerprint information extraction.

Algorithm	BSLD	ACE	CEM	MF
AUC	0.9747	0.9706	0.9426	0.9697

TABLE 2: Evaluation of AUC of tumor area information extraction.

Algorithm	BSLD	ACE	CEM	MF
AUC	0.9910	0.9758	0.9905	0.9828

around the location of the body, the effects in the medical application field are not significant. We need to define a higher decision threshold to get accurate images if we try to get a more accurate image. The ROC curve is plotted according to the ROI information given by the control data as shown in Figure 9. We could find that BSLD method is still superior to the other three methods. AUC in Table 2 is also improved

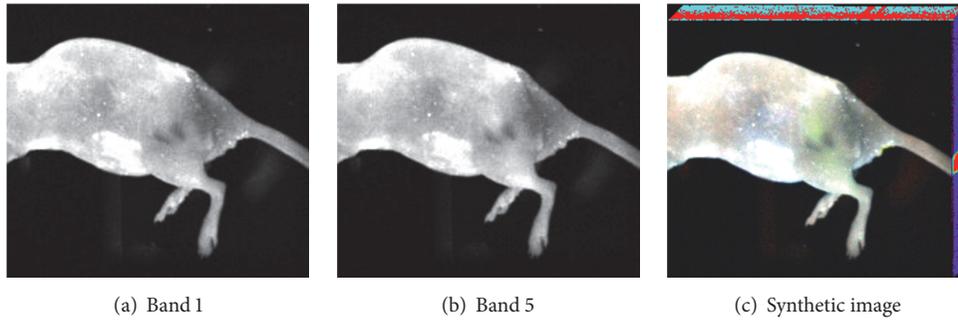


FIGURE 7: The four bands in the tumor data and the synthetic image.

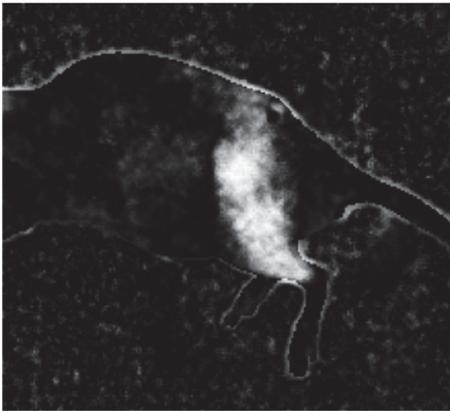


FIGURE 8: The effect of tumor area information extraction.

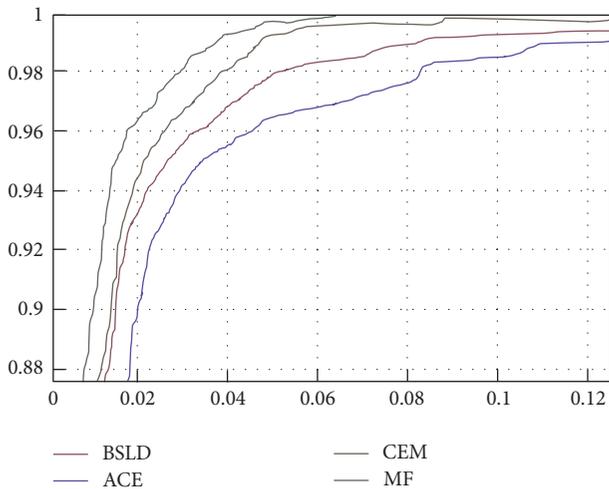


FIGURE 9: ROC curve evaluation of tumor area information extraction.

slightly, and its value reaches to 99.1%. So we can conclude that BSLD could accurately determine the tumor area.

4. Conclusion

In this paper, a new method of hyperspectral image biometric information extraction is proposed, which can make up for

the deficiency of background hyperspectral image processing method so that background information is suppressed and target information is easier to be extracted. The two groups of experiments show that this method can extract the target biological information completely and clearly, and the extraction effect is superior to the classical ACE, CEM, and MF methods.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work is supported by the CRSRI Open Research Program (Program no. CKWV2016380/KY).

References

- [1] J. Wu, S. Pan, X. Zhu, C. Zhang, and X. Wu, "Multi-instance learning with discriminative bag mapping," *IEEE Transactions on Knowledge and Data Engineering*, pp. 1-1, 2018.
- [2] J. Wu, S. Pan, X. Zhu, C. Zhang, and P. S. Yu, "Multiple Structure-View Learning for Graph Classification," *IEEE Transactions on Neural Networks and Learning Systems*, vol. PP, no. 99, pp. 1-16, 2017.
- [3] J. Wu, S. Pan, X. Zhu, C. Zhang, and X. Wu, "Positive and Unlabeled Multi-Graph Learning," *IEEE Transactions on Cybernetics*, vol. 47, no. 4, pp. 818-829, 2016.
- [4] B. Du, W. Xiong, J. Wu, L. Zhang, L. Zhang, and D. Tao, "Stacked Convolutional Denoising Auto-Encoders for Feature Representation," *IEEE Transactions on Cybernetics*, vol. 47, no. 4, pp. 1017-1027, 2017.
- [5] S. A. Sudiro, M. Paindavoine, and T. M. Kusuma, "Simple fingerprint minutiae extraction algorithm using crossing number on valley structure," in *Proceedings of the 2007 IEEE Workshop on Automatic Identification Advanced Technologies (AUTOID '07)*, pp. 41-44, June 2007.
- [6] C.-J. Lee and S.-D. Wang, "Fingerprint feature extraction using Gabor filters," *IEEE Electronics Letters*, vol. 35, no. 4, pp. 288-290, 1999.
- [7] Q. Zhang and H. Yan, "Fingerprint classification based on extraction and analysis of singularities and pseudo ridges," *Pattern Recognition*, vol. 37, no. 11, pp. 2233-2243, 2004.
- [8] Q. Abbas, M. E. Celebi, and I. F. García, "Skin tumor area extraction using an improved dynamic programming approach," *Skin Research and Technology*, vol. 18, no. 2, pp. 133-142, 2012.

- [9] J. Gil, H. Wu, and B. Y. Wang, "Image analysis and morphometry in the diagnosis of breast cancer," *Microscopy Research and Technique*, vol. 59, no. 2, pp. 109–118, 2002.
- [10] B. Du, Y. Zhang, L. Zhang, and D. Tao, "Beyond the sparsity-based target detector: a hybrid sparsity and statistics-based detector for hyperspectral images," *IEEE Transactions on Image Processing*, vol. 25, no. 11, pp. 5345–5357, 2016.
- [11] B. Du, M. Zhang, L. Zhang, R. Hu, and D. Tao, "PLTD: Patch-Based Low-Rank Tensor Decomposition for Hyperspectral Images," *IEEE Transactions on Multimedia*, vol. 19, no. 1, pp. 67–79, 2017.
- [12] D. Manolakis and G. Shaw, "Detection algorithms for hyperspectral imaging applications," *IEEE Signal Processing Magazine*, vol. 19, no. 1, pp. 29–43.
- [13] J. Harsanyi C, H. Farrand W, and I. Chang C, "Detection of subpixel signatures in hyperspectral image sequences," in *Proceedings of the American Society for Photogrammetry and Remote Sensing*, pp. 236–247, 1994.
- [14] L. L. Scharf and L. T. McWhorter, "Adaptive matched subspace detectors and adaptive coherence estimators," in *Proceedings of the 1996 30th Asilomar Conference on Signals, Systems & Computers. Part 2 (of 2)*, pp. 1114–1117, November 1996.
- [15] B. Park, W. R. Windham, K. C. Lawrence, and D. P. Smith, "Contaminant Classification of Poultry Hyperspectral Imagery using a Spectral Angle Mapper Algorithm," *Biosystems Engineering*, vol. 96, no. 3, pp. 323–333, 2007.
- [16] C.-I. Chang, "Orthogonal Subspace Projection (OSP) revisited: a comprehensive study and analysis," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 43, no. 3, pp. 502–518, 2005.
- [17] K. Liu, L.-f. Zhang, H. Yang et al., "Study on hyperspectral detection of unstructured background targets based on object-oriented analysis," *Spectroscopy and Spectral Analysis*, vol. 33, no. 6, 2013.
- [18] B. Du, Z. Wang, L. Zhang et al., "Exploring Representativeness and Informativeness for Active Learning," *IEEE Transactions on Cybernetics*, vol. 47, no. 1, pp. 14–26, 2017.
- [19] S. Matteoli, N. Acito, M. Diani, and G. Corsini, "An automatic approach to adaptive local background estimation and suppression in hyperspectral target detection," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 49, no. 2, pp. 790–800, 2011.
- [20] J. M. Bioucas-Dias and J. M. P. Nascimento, "Hyperspectral subspace identification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 46, no. 8, pp. 2435–2445, 2008.
- [21] L. Zhang, L. Zhang, D. Tao, X. Huang, and B. Du, "Compression of hyperspectral remote sensing images by tensor approach," *Neurocomputing*, vol. 147, no. 1, pp. 358–363, 2015.
- [22] E. R. DeLong, D. M. DeLong, and D. L. Clarke-Pearson, "Comparing the areas under two or more correlated receiver operating characteristic curves: a nonparametric approach," *Biometrics*, vol. 44, no. 3, pp. 837–845, 1988.
- [23] B. Du and L. Zhang, "A discriminative metric learning based anomaly detection method," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 52, no. 11, pp. 6844–6857, 2014.

Research Article

Unsupervised Domain Adaptation Using Exemplar-SVMs with Adaptation Regularization

Yiwei He ¹, Yingjie Tian ^{2,3,4}, Jingjing Tang ⁵ and Yue Ma ⁵

¹School of Computer and Control Engineering, University of Chinese Academy of Sciences, Beijing 100049, China

²Research Center on Fictitious Economy and Data Science, Chinese Academy of Sciences, Beijing 100190, China

³Key Laboratory of Big Data Mining and Knowledge Management, Chinese Academy of Sciences, Beijing 100190, China

⁴School of Economics and Management, University of Chinese Academy of Sciences, Beijing 100190, China

⁵School of Mathematical Sciences, University of Chinese Academy of Sciences, Beijing 100049, China

Correspondence should be addressed to Yingjie Tian; tyj@ucas.edu.cn

Received 28 August 2017; Accepted 18 February 2018; Published 22 April 2018

Academic Editor: Shirui Pan

Copyright © 2018 Yiwei He et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Domain adaptation has recently attracted attention for visual recognition. It assumes that source and target domain data are drawn from the same feature space but different margin distributions and its motivation is to utilize the source domain instances to assist in training a robust classifier for target domain tasks. Previous studies always focus on reducing the distribution mismatch across domains. However, in many real-world applications, there also exist problems of sample selection bias among instances in a domain; this would reduce the generalization performance of learners. To address this issue, we propose a novel model named Domain Adaptation Exemplar Support Vector Machines (DAESVMs) based on exemplar support vector machines (exemplar-SVMs). Our approach aims to address the problems of sample selection bias and domain adaptation simultaneously. Comparing with usual domain adaptation problems, we go a step further in slacking the assumption of i.i.d. First, we formulate the DAESVMs training classifiers with reducing Maximum Mean Discrepancy (MMD) among domains by mapping data into a latent space and preserving properties of original data, and then, we integrate classifiers to make a prediction for target domain instances. Our experiments were conducted on Office and Caltech10 datasets and verify the effectiveness of the model we proposed.

1. Introduction

Over the past decades, machine learning technologies have achieved significant success in various areas, such as computer vision [1], natural language processing [2], and video detection [3]. However, traditional machine learning methods assume that training and testing data come from the same domain, which implies that training or testing data are drawn from the same distribution and represented in the same feature spaces. This assumption is too violated to be held in the real world as collecting suitable and enough labeled data is time consuming and an expensive manual effort. Lacking labeled data, most of traditional machine learning methods always lose their generalization performance in reality. Therefore, it is desired to utilize the data of the relational domain to help training a robust learner for target domains. Driven by this requirement, transfer learning has

rapidly developed in recent years [4]. Transfer learning slacks the assumption of the traditional machine learning in which data or labels are drawn from the same distribution and represented in the same feature space. In the transfer learning settings, it is always assumed that domains are similar or related, with even no relationships, which is instead of i.i.d. assumption. Thus, transfer learning has a strong motivation when developing the classical machine learning functions or applying the functions to real-world applications. Besides, transfer learning can be regarded as a supplement of classical machine learning methods. One is the problem of covariate shift or sample selection bias. Another motivation is that we want to train a universal or general model as a predictor for all the tasks, viewed as the parameter or learner shared. It is also considered as a goal of Artificial General Intelligence. Transfer learning aims to utilize source or related domains to help target domain tasks. It has achieved significant success

in various practical applications, such as face recognition [5], natural language processing [6], cross-language text classification [7], WiFi localization [8], or medicine image [9].

Domain adaptation is a subproblem of transfer learning which assumes that source and target domain data are generated from the same feature and label space but different margin probability distributions. It aims to solve the problems that there is none or less labeled data in the target domain and usually use labeled data in the source domain to assist the training of target domain tasks. Massive works focus on the domain adaptation problems, and they also extend to some applications, such as WiFi location, text sentiment analysis, and image classification for multidomains. Since distribution mismatch generally exists in the real-world applications, there is also some other research area concern about domain adaptation. For example, extreme learning machine (ELM) is an efficient model for training single-hidden layer networks [10]. There are also some ELM works in a domain adaptation setting [11, 12]. They utilize most previous domain adaptation classifiers that have added constraint term which is based on using instance reweighting to minimize Maximum Mean Discrepancy (MMD) [13]. However, these methods need to assume that the difference between the source and target domain is not too large. Namely, this idea requires that different domains are similar.

Most pattern recognition problems can be transformed into several basic classification tasks. Generally speaking, classification tasks assume that a category can be represented by a hyperplane [14, 15], and most of the machine learning algorithms aim to learn hyperplanes to predict for unseen instances. Meanwhile, to improve the ability of representation by a hyperplane, there are some works which cluster the samples first and then solve the classification tasks on the clusters. In contrast to the category classification tasks, a cluster classifier can include more information about the positive category, but the more risks of overfitting. Motivated by the object detection, [16] proposed an extreme classification model training the classifiers for every positive instance and all the negative instances named exemplar support vector machines (E-SVMs). In fact, exemplar-SVMs can be viewed as an extreme situation of cluster-level SVM, in which every positive sample is regarded as a cluster. There are two viewpoints about the reason why the exemplar-SVM achieves a surprising generalization performance. One of the viewpoints is taking the exemplar-SVMs as a representation with complete details of positive instances. In other words, every classifier captures details of the positive instance like background, corner, color, or orientations and most of the classifiers can describe the category more intrinsically. From transfer learning viewpoint, training data cannot satisfy the underlying assumption of i.i.d., as every instance in the training set may be different from each other, namely, sample selection bias [17]. Each exemplar-SVMs classifier is trained on a high weight positive sample and other negative samples; it can represent the positive sample well in the same distribution. Recently, [18] extends exemplar-SVMs into a transfer learning form which uses loss function reweighting and adds a low-rank regularization item for classifiers.

In this work, we propose a novel model to address unsupervised domain adaptation problems that there is no label on target domain data. Furthermore, it permits distribution mismatch among instances. In our model, we train kernel exemplar classifiers for every positive instance and then integrate the classifier to make a prediction for target domain data. To align the distribution mismatch, we embed the regularization item based on TCA in our classifiers. In our opinion, the model constructs the bridge to transfer the knowledge, and we use the information in the kernel matrix which includes the instances representation in the high-dimension space to assist classifier training across domains. For the problem of sample selection bias, we integrate the classifiers to make a prediction. Basically, the step of integration is to expand the representation of hyperplanes that entirely take advantage of details learned before.

Our contributions are as follows. (1) We propose a novel unsupervised domain adaptation model based on exemplar-SVMs named Domain Adaptation Exemplar Support Vector Machines (DAESVMs), and it improves standard domain adaptation prediction accuracy by transferring knowledge across domains. (2) Every DAESVM classifier constructs a bridge that transmits knowledge from the source domain to target domain. Compared with the traditional two-step method, this strategy thoroughly searches the optimization point of the model which makes the classification hyperplane more precious about domains. (3) To solve the problem of sample selection bias, we use the ensemble methods to integrate the classifiers. The process of the ensemble is similar to slacking the classification hyperplane, which drops off some unreliable classification results and use the reliable parts to make a prediction. (4) We bring in the method of the pseudo label in DAESVMs inspired by [19] to supplement the information of target domain, and the experiments verify the effectiveness of the pseudo label. (5) We push a step further to extend to implementing DAESVMs on the multidomain adaptation. The rest of this paper is organized as follows. In Section, we introduce the notation of the problem. Meanwhile, we review the related works of domain adaptation, exemplar-SVM, and Transfer Component Analysis (TCA). In Section, we introduce the deduction process of DAESVM and formulate the model. In Section, we propose the optimization algorithm for our model. In Section, we integrate all the DAESVMs classifiers to make a prediction. In Section, we analyze the experiments on some transfer learning dataset to verify the effectiveness of DAESVMs. In Section, we conclude our work and give an expectation.

2. Notation and Related Works

This section will introduce the notation and related works about this paper.

2.1. Notation. In this paper, we use the notation of [4] definition in transfer learning, and the definition just considers the condition of one source domain and one target domain. First, it needs to define the *Domain* and *Task*. Domain \mathcal{D} is composed of a feature space \mathcal{X} and a margin probability distribution $P(x)$, namely, $\mathcal{D} = \{\mathcal{X}, P(x)\}$, $x \in \mathcal{X}$. Task \mathcal{T}

is composed of a label space \mathcal{Y} and a prediction model $f(x)$, namely, $\mathcal{T} = \{\mathcal{Y}, f(x)\}$, $y \in \mathcal{Y}$. From view of probability, $f(x) = P(y | x)$. Notations in this paper which are frequently used are summarized in the Notations and Descriptions section. The definition of transfer learning is as follows: Give a source domain data $\mathcal{D}_S = \{(x_{S_1}, y_{S_1}), \dots, (x_{S_{n_S}}, y_{S_{n_S}})\}$ and a source task \mathcal{T}_S and a target domain data which is unlabeled $\mathcal{D}_T = \{(x_{T_1}), \dots, (x_{T_{n_T}})\}$ and a target task \mathcal{T}_T . Transfer learning aims to utilize \mathcal{D}_S and \mathcal{D}_T to help train a robust prediction model $f_t(x)$ on the condition of $\mathcal{D}_S \neq \mathcal{D}_T$ or $\mathcal{T}_S \neq \mathcal{T}_T$.

2.2. Domain Adaptation. As a subproblem of transfer learning, domain adaptation has achieved great success and is utilized in many applications. It assumes that source and target domain data have the same feature space, label space, and prediction function, from the view of probability, equaling conditional probabilities distribution, namely, $f_S(x) \neq f_T(x)$ or $P_S(y | x) \neq P_T(y | x)$. It is agreed that the approaches of domain adaptation can be divided into three parts, reweighting approach, feature transfer approach, and parameter shared approach.

(1) Reweighting Approaches. In the transfer learning tasks, the basic idea of utilizing the source data to help training target predictor is to reduce the discrepancy between the source and target data as far as possible. Under the assumption that source and target domains have a lot of overlapping features, a conventional method is reweighting or selecting the source domain instances to correct the marginal probability distribution mismatch. Based on the metric distance method between distributions named Maximum Mean Discrepancy (MMD), [20] proposed a technique called Kernel Mean Minimum (KMM) revising the weight of every instance to minimize MMD between the source and target domain. Being similar to KMM, [21] used the same idea but a different metric method to adjust the discrepancy of domains. Reference [22] used the strategy of AdaBoost to update the weights of source domain data, which improved the weight of instances in favor of classification task. It also introduced the generalization error bounds of model based on the PAC learning theory. In recent years, [23] used a two-step approach; first is sampling the instances which are similar with other domains as landmarks, and then use these landmarks to map the data into a high-dimension space, after which it is more overlapping. Reference [24] solved the same problem but slacked the similarity assumption; it assumes that there are no relationships between the source and target domain. The model named Selective Transfer Machine (STM) reweights the instance of personal faces to train a generic classifier. Most of instance-based transfer learning techniques use KMM to measure the difference of the distributions, and these methods are applied in many areas, such as facial action unit detection [25] and prostate cancer mapping [26].

(2) Feature Transfer Approaches. Compared with instance-based approaches, feature-based approaches slack the similarity assumption. It assumes that source and target domain

share some features named shared features, and domains have their own features named spec-features [27]. For example, when we train a task that uses movie critical to help sofa critical sentiment analysis classification task. The word “comfortable” is always nonzero in the sofa domain features but always zero in the movie domain features. This word is the spec-feature of sofa domain feature. Feature transfer approaches aim to find a shared latent subspace where the distance between the source and target domain is minimized. Reference [28] proposed an unsupervised domain adaptation approach named Geodesic Flow Kernel (GFK) based on kernel method. GFK maps data into Grassmann manifolds and constructs geodesic flows to reduce the mismatch among domains. It effectively exploits intrinsic low-dimensional structures of data in domains. To solve problems of cross-domain natural language processing (NLP), [29] proposed a general method structural correspondence learning (SCL) to learn a discriminative predictor by identifying correspondences from features in domains. Primarily, SCL finds the pivot features and then links the shared features with each other. Reference [7] learned a predictor by mapping the target kernel matrix to a submatrix of the source kernel matrix. The deep neural network is used not for learning essential features but also for domain adaptation. Reference [30] proposed a neural network architecture for domain adaptation named Deep Adaptation Network (DAN) and extended it to joint adaptation networks (JAN) [31]. Reference [32] discussed the transferable domain features on the deep neural network.

(3) Parameter-Based Approaches. The core idea of parameter-based approaches aims to transfer parameters from source to target domain tasks. It assumes that different domains share some parameters and these parameters could be utilized for domains. Reference [33] proposed Adaptive Support Vector Machine (A-SVM) as a general method to adopt new domains. A-SVM trains an auxiliary classifier firstly and then learns the target predictor based on the original parameters. Reference [34] reweighted prediction of the source classifier on target domain by signing distance between domains.

2.3. Exemplar Support Vector Machines. Reference [16] is proposed for object detection and getting high performance. It trains classifiers on every positive instance from all negative instances. Every positive instance is an exemplar and the classifier corresponding to it can be viewed as a representation of the positive instance. In the process of the prediction, every classifier predicts a value for the test instance and uses a function to make a calibration for the value and then gets the high score classifiers result as a predicted class. The exemplar-SVMs solve the problem that a hyperplane is hard to represent a category instance and utilize an extreme strategy to train predictor. In [35], they gather the training procession into one model and enter the nuclear norm regularization to the scene of domain generalization which assumes target domain is unseen. They also extend the model to the problem of domain generalization and multiview [36, 37]. In [38], they reduced two hyperparameters into one and spread exemplar-SVMs to a kernel form.

2.4. Transfer Component Analysis. Reference [39] proposed a dimension reduction method called maximum mean discrepancy embedding (MMDE). By minimizing the distance of source and target domain data distribution in a shared latent space, the source domain data is utilized to assist training classifier on the target domain. MMDE is not only to minimize the distance between the domains in the latent space but also preserve the properties of data by maximum of the variance of data. Based on the MMDE, [40] extended it to have the ability of deal with the unseen instance and reduce the computation complexity of MMDE. Substantially, TCA simplifies the process of learning kernel matrix instead by transforming init kernel matrix. The optimization of this problem is equal to a solution in m leading eigenvectors of object matrix.

3. Domain Adaptation Exemplar Support Vector Machine

In this section, we present the formulation of Domain Adaptation Exemplar Support Vector Machine (DAESVM). In the remainder of this paper, we use a lowercase letter in boldface to represent a column vector and an uppercase in boldface to represent a matrix. The notation mentioned in Section is extended. We use $\mathbf{x}_i^+, i \in \{1, \dots, n_s^+\}$, where n_s^+ is the number of positive instances, to represent a positive instance, and $\mathbf{x}_j^-, j \in \{1, \dots, n_s^-\}$, where n_s^- is the number of negative instances, to represent a negative instance. The set of negative samples are written as N^- . This section introduces the formulation procession of an exemplar classifier. In fact, we need to train exemplar classifiers in the number of source domain instances and the method which integrates these classifiers is proposed in Section.

3.1. Exemplar-SVM. The exemplar-SVM is constructed by an extreme idea of training a classifier by a positive instance from all the negative instances and then calibrating the outputs of classifiers into a probability distribution to separate the samples. The model trains the number of positive instance classifiers. Learning a classifier which aims to separate a positive instance from all the negative instance can be modeled as

$$f(\mathbf{w}, b) = \|\mathbf{w}\|^2 + C_1 h(\mathbf{w}^T \mathbf{x}^+ + b) + C_2 \sum_{\mathbf{x}_i^- \in n_s^-} h(-\mathbf{w}^T \mathbf{x}_i^- - b), \quad (1)$$

where $\|\cdot\|$ is 2-norm of a vector and C_1 and C_2 are the tradeoff parameters corresponding to C in SVM for balancing the positive and negative error cost. $h(x) = \max(0, 1 - x)$ is a hinge loss function.

The formulation (1) is the primal problem of exemplar-SVM, and we can find the dual problem for utilizing kernel method. The dual formulation can be written as follows [38]:

$$\begin{aligned} \min_{\alpha} \quad & \alpha^T \tilde{\mathbf{K}} \alpha - \mathbf{e}^T \alpha, \\ \text{s.t.} \quad & \alpha_0 - \sum_{i=1}^{n_s^-} \alpha_i = 0, \end{aligned}$$

$$0 \leq \alpha_0 \leq C_1,$$

$$0 \leq \alpha_i \leq C_2,$$

$$\forall i \geq 1.$$

(2)

$\alpha = (\alpha_0, \alpha_1, \dots, \alpha_{n_s^-}) \in \mathcal{R}^{n_s^-+1}$ are Lagrangian multipliers. \mathbf{e} is an identity vector. We take this model as an exemplar learner. The matrix $\tilde{\mathbf{K}} \in \mathcal{R}^{(n_s^-+1) \times (n_s^-+1)}$ is composed of

$$\tilde{\mathbf{K}} = \begin{bmatrix} k(\mathbf{x}^+, \mathbf{x}^+) & -\mathbf{k}^T \\ -\mathbf{k} & \mathbf{K} \end{bmatrix} \in \mathcal{R}^{(n_s^-+1) \times (n_s^-+1)}$$

$$k \in \mathcal{R}^{N_s^-}, \quad (3)$$

$$k_i = k(\mathbf{x}, \mathbf{x}_i^-),$$

$$\mathbf{K}_{ij} = k(\mathbf{x}_i^-, \mathbf{x}_j^-).$$

3.2. Pseudo Label for Kernel Matrix. To make the best use of samples in source or target, we construct the kernel matrix on both domain data. However, in the dual problem of SVM, kernel matrix \mathbf{K} needs to be supplied labeled data. Our model is based on the unsupervised domain adaptation problem in which only source domain data are labeled. Motivated by [19], we use the pseudo label to help model training. Pseudo labels are predicted by classical classifiers, SVM in our model, which train on the source labeled data. Due to the distribution mismatch between source and target domain, there may be many labels incorrect. Followed by [19], we assume that the pseudo class centroids calculated by them may reside not far apart from the true class centroids. Thus, we use both domain data to supplement the kernel matrix \mathbf{K} with label information. In our experiments, we testify this method is effective.

3.3. Exemplar Learner in Domain Adaptation Form. In fact, each exemplar learner is an SVM in kernel form which is trained by a positive instance and all the negative instances. In the opinion of [16], a discriminative exemplar classifier can be taken as a representation of a positive instance. However, in the task of object detection or image classification, this parametric form representation is feasible because of some characteristics in samples, such as angle, color, orientations, and background, which are hard to represent. The instance-based parametric discriminative classifier can include more information about positive samples. Similarly, with the motivation of transfer learning, we can view a positive instance as a domain, and there is some mismatch among domains. Our model aims to correct this mismatch and reduce the distance from the target domain. We construct an exemplar learner distance metric of domains from MMD and it can be written as

$$\begin{aligned} \text{dist}(\mathbf{x}_S, \mathbf{x}_T) \\ = \left\| \phi(\mathbf{x}_S^+) + \frac{1}{n_s^-} \sum_{i=1}^{n_s^-} \phi(\mathbf{x}_S^-) - \frac{2}{n_T} \sum_{i=1}^{n_T} \phi(\mathbf{x}_T) \right\|_{\mathcal{H}}^2. \end{aligned} \quad (4)$$

However, it is just a metric of distance which is satisfied with our requirement of minimizing this distance by some transformation. Motivated by Transfer Component Analysis (TCA), we want to map the instance into a latent space that the instances from source and target domain are more similar and assume this mapping is $P(x)$. Namely, we aim to minimize MMD distance between domains by mapping instances into another space. We extend the distance function as follows:

$$\begin{aligned} \text{dist}(\mathbf{x}_S, \mathbf{x}_T) = & \left\| \phi(P(\mathbf{x}_S^+)) + \frac{1}{n_S} \sum_{i=1}^{n_S} \phi(P(\mathbf{x}_S^-)) \right. \\ & \left. - \frac{2}{n_T} \sum_{i=1}^{n_T} \phi(P(\mathbf{x}_T)) \right\|_{\mathcal{H}}^2. \end{aligned} \quad (5)$$

Corresponding to a general approach, it always reformulates (4) to construct a kernel matrix form. We define the Gram matrices on the source positive domain, source negative domain, and target domain. The kernel matrix \mathbf{K} is composed of nine submatrices, $\mathbf{K}_{T,+}$, $\mathbf{K}_{T,-}$, $\mathbf{K}_{T,T}$, $\mathbf{K}_{+,+}$, $\mathbf{K}_{-,-}$, $\mathbf{K}_{+,-}$, $\mathbf{K}_{+,S}$, $\mathbf{K}_{-,T}$, $\mathbf{K}_{-,+}$ where $\mathbf{K} = [\phi(x_i)^T \phi(x_j)]$.

$$\mathbf{K} = \begin{bmatrix} \mathbf{K}_{+,+} & \mathbf{K}_{+,-} & \mathbf{K}_{+,T} \\ \mathbf{K}_{-,-} & \mathbf{K}_{-,-} & \mathbf{K}_{-,T} \\ \mathbf{K}_{T,+} & \mathbf{K}_{T,-} & \mathbf{K}_{T,T} \end{bmatrix} \in \mathcal{R}^{(1+n_S^+ + n_T) \times (1+n_S^- + n_T)}, \quad (6)$$

and it constructs the coefficient matrix \mathbf{L} ,

$$\mathbf{L}_{ij} = \begin{cases} 1, & \text{if } \mathbf{x}_i, \mathbf{x}_j \in \mathbf{X}_S^+, \\ \frac{1}{n_S}, & \text{if } \mathbf{x}_i \in \mathbf{X}_S^+, \mathbf{x}_j \in \mathbf{X}_S^-, \\ -\frac{2}{n_T}, & \text{if } \mathbf{x}_i \in \mathbf{X}_S^+, \mathbf{x}_j \in \mathbf{X}_T, \\ -\frac{2}{n_S n_T}, & \text{if } \mathbf{x}_i \in \mathbf{X}_T, \mathbf{x}_j \in \mathbf{X}_S^-, \\ \frac{1}{n_S}, & \text{if } \mathbf{x}_i, \mathbf{x}_j \in \mathbf{X}_S^-, \\ \frac{4}{n_T^2}, & \text{if } \mathbf{x}_i, \mathbf{x}_j \in \mathbf{X}_T. \end{cases} \quad (7)$$

Thus, the primal distance function is represented by \mathbf{KL} . Motivated by TCA [40], the mapping for primal data is equal to the transformation of kernel matrix generated by the source and target domain data. Utilizing the low-dimension transform matrix $\widetilde{\mathbf{M}} \in \mathcal{R}^{(1+n_S^+ + n_T) \times m}$ reduces the dimension of the primal kernel matrix. It maps the empirical kernel map $\mathbf{K} = (\mathbf{K}\mathbf{K}^{-1/2})(\mathbf{K}^{-1/2}\mathbf{K})$ into an m -dimensional shared space. Mostly, we replaced the distance function \mathbf{KL} by $(\mathbf{KMM}^T\mathbf{KL})$. In our case, we follow [40] and minimize the trace of the distance,

$$\text{dist}(\mathbf{x}_S^+, \mathbf{x}_S^-, \mathbf{x}_T) = \text{tr}(\mathbf{M}^T \mathbf{KLKM}). \quad (8)$$

For controlling the complexity of \mathbf{M} and preserving the data characteristic, we add the regularization and constraint item.

The domain adaptation item is formulated followed from TCA and written as

$$\begin{aligned} \Omega(\mathbf{x}_S^+, \mathbf{x}_S^-, \mathbf{x}_T) = & \text{tr}(\mathbf{M}^T \mathbf{KLKM}) + \mu \text{tr}(\mathbf{M}^T \mathbf{M}) \\ \text{s.t. } & \mathbf{M}^T \mathbf{KHKM} = \mathbf{I}_m, \end{aligned} \quad (9)$$

where $\mu > 0$ is a tradeoff parameter and $\mathbf{I}_m \in \mathcal{R}^{(m \times m)}$ is an identity matrix. $\mathbf{H} = \mathbf{I}_{n_S^- + n_T + 1} - (1/(n_S^- + n_T + 1))\mathbf{e}\mathbf{e}^T$ is a centering matrix.

Furthermore, the objective function of dual SVM needs to be added to the training label information which is similar to our model. Thus, we construct the training label matrix \mathbf{U}

$$\mathbf{U} = \text{diag}(\mathbf{y}_S^+, \mathbf{y}_S^-, \mathbf{y}_T). \quad (10)$$

\mathbf{y}_S^+ is the label of a positive instance, \mathbf{y}_S^- is the label vector of negative source instances, and \mathbf{y}_T is the pseudo labels of target instances which are predicted by SVM before. It can be rewritten in another form:

$$\mathbf{U} = \text{diag} \left(\underbrace{1, -1, \dots, -1}_{n_S^-}, \underbrace{\mathbf{y}_T^1, \dots, \mathbf{y}_T^{n_T}}_{n_T} \right). \quad (11)$$

Label matrix \mathbf{U} provides the information of source domain data labels and target domain pseudo labels. The matrix $\widetilde{\mathbf{K}}$ in a dual problem of exemplar-SVM (2) is primal data kernel matrix. We want to replace it by mapping the kernel matrix into a latent subspace. Namely, replace \mathbf{K} by $\widetilde{\mathbf{K}}$ and the final objective function of each DAESVM model is formulated as follows:

$$\begin{aligned} \min_{\alpha, \mathbf{M}} & \alpha^T \widetilde{\mathbf{K}} \alpha - \mathbf{e}^T \alpha + \lambda \text{tr}(\mathbf{M}^T \mathbf{KLKM}) \\ & + \mu \text{tr}(\mathbf{M}^T \mathbf{M}), \\ \text{s.t. } & \alpha_0 - \sum_{i=1}^{n_S^- + n_T} \alpha_i = 0, \\ & 0 \leq \alpha_0 \leq C_1, \\ & 0 \leq \alpha_i \leq C_2, \\ & \forall i \geq 1, \\ & \mathbf{M}^T \mathbf{KHKM} = \mathbf{I}_m, \\ & \widetilde{\mathbf{K}} = \mathbf{UKMM}^T \mathbf{KU}. \end{aligned} \quad (12)$$

4. Optimization Algorithm

To minimize problem (12), we adopt the alternated optimization method which alternates between solving two subproblems over parameter α and mapping matrix \mathbf{M} . Under these methods, the alternated optimization approach is guaranteed to decrease the objective function. Algorithm 1 summarizes the optimization procedure of problem (12) which we formulated.

Input: $\mathbf{X}^{tr}, \mathbf{X}^{te}$; parameter λ, μ, m, C_1 and C_2 ;
Output: optimal α and \mathbf{M}

- (1) initial $\alpha = \mathbf{0}$;
- (2) Construct kernel matrix \mathbf{K} from \mathbf{X}^{tr} and \mathbf{X}^{te} based on (6); coefficient matrix \mathbf{L} based on (7); centering matrix \mathbf{H} ; label matrix \mathbf{U} based on (11).
- (3) **repeat**
- (4) Update transformation matrix \mathbf{M} when fix α
- (5) Eigendecompose the optimization matrix of $(\mathbf{K}\mathbf{U}\alpha\alpha^T\mathbf{U}\mathbf{K} + \lambda\mathbf{K}\mathbf{L}\mathbf{K} - \mu\mathbf{I}_m)^{-1}\mathbf{K}\mathbf{H}\mathbf{K}$ and select m leading eigenvectors to construct the transformation matrix \mathbf{M}
- (6) Solve the convex optimization problem for fixed \mathbf{M} to optimize α
- (7) **until** Convergence

ALGORITHM 1: Domain Adaptation Exemplar Support Vector Machine.

Minimizing over α . The optimization over α can be rewritten into the following form:

$$\begin{aligned} \min_{\mathbf{M}} \quad & \alpha^T \mathbf{U}\mathbf{K}\mathbf{M}\mathbf{M}^T \mathbf{K}\mathbf{U}\alpha - \mathbf{e}^T \alpha + \lambda \operatorname{tr}(\mathbf{M}^T \mathbf{K}\mathbf{L}\mathbf{K}\mathbf{M}) \\ & + \mu \operatorname{tr}(\mathbf{M}^T \mathbf{M}), \\ \text{s.t.} \quad & \mathbf{M}^T \mathbf{K}\mathbf{H}\mathbf{K}\mathbf{M} = \mathbf{I}_m. \end{aligned} \quad (13)$$

Being similar to TCA, the formulation is containing a non-convex norm constraint, and we transform this optimization problem by reformulating as

$$\max_{\mathbf{M}} \quad \operatorname{tr}\left(\left(\mathbf{M}^T (\mathbf{K}\mathbf{U}\alpha\alpha^T \mathbf{U}\mathbf{K} + \lambda\mathbf{K}\mathbf{L}\mathbf{K} - \mu\mathbf{I}_m) \mathbf{M}\right)^{-1} \mathbf{M}^T \mathbf{K}\mathbf{H}\mathbf{K}\mathbf{M}\right). \quad (14)$$

Proof. The Lagrangian of (12) is

$$\begin{aligned} \mathcal{L}(\mathbf{M}, \mathbf{Z}) = & \alpha^T \mathbf{U}\mathbf{K}\mathbf{M}\mathbf{M}^T \mathbf{K}\mathbf{U}\alpha - \mathbf{e}^T \alpha \\ & + \lambda \operatorname{tr}(\mathbf{M}^T \mathbf{K}\mathbf{L}\mathbf{K}\mathbf{M}) - \mu \operatorname{tr}(\mathbf{M}^T \mathbf{M}) \\ & - \operatorname{tr}\left(\left(\mathbf{M}^T \mathbf{K}\mathbf{H}\mathbf{K}\mathbf{M} - \mathbf{I}_m\right) \mathbf{Z}\right). \end{aligned} \quad (15)$$

Because the initial kernel matrix \mathbf{K} is a symmetric matrix and we can rewrite the first term of (15),

$$\begin{aligned} \alpha^T \mathbf{U}\mathbf{K}\mathbf{M}\mathbf{M}^T \mathbf{K}\mathbf{U}\alpha &= \operatorname{tr}\left(\alpha^T \mathbf{U}\mathbf{K}\mathbf{M}\mathbf{M}^T \mathbf{K}\mathbf{U}\alpha\right) \\ &= \operatorname{tr}\left[\left(\mathbf{M}^T \mathbf{K}^T \mathbf{U}\alpha\right)^T \left(\mathbf{M}^T \mathbf{K}^T \mathbf{U}\alpha\right)\right] \\ &= \operatorname{tr}\left[\left(\mathbf{M}^T \mathbf{K}^T \mathbf{U}\alpha\right) \left(\mathbf{M}^T \mathbf{K}^T \mathbf{U}\alpha\right)^T\right] \\ &= \operatorname{tr}\left(\mathbf{M}^T \mathbf{K}\mathbf{U}\alpha\alpha^T \mathbf{U}\mathbf{K}\mathbf{M}\right). \end{aligned} \quad (16)$$

The original Lagrangian formulation is written as follows:

$$\begin{aligned} & \operatorname{tr}\left(\mathbf{M}^T (\mathbf{K}\mathbf{U}\alpha\alpha^T \mathbf{U}\mathbf{K} + \lambda\mathbf{K}\mathbf{L}\mathbf{K} - \mu\mathbf{I}_m) \mathbf{M}\right) \\ & - \operatorname{tr}\left(\left(\mathbf{M}^T \mathbf{K}\mathbf{H}\mathbf{K}\mathbf{M} - \mathbf{I}_m\right) \mathbf{Z}\right). \end{aligned} \quad (17)$$

The derivative of (17) is

$$\left(\mathbf{K}\mathbf{U}\alpha\alpha^T \mathbf{U}\mathbf{K} + \lambda\mathbf{K}\mathbf{L}\mathbf{K} - \mu\mathbf{I}_m\right) \mathbf{M} - \mathbf{K}\mathbf{H}\mathbf{K}\mathbf{M}\mathbf{Z}. \quad (18)$$

We set the derivative above to zero, and we get \mathbf{Z} as

$$\mathbf{Z} = \left(\mathbf{M}^T \mathbf{K}\mathbf{H}\mathbf{K}\mathbf{M}\right)^\dagger \mathbf{M}^T \left(\mathbf{K}\mathbf{U}\alpha\alpha^T \mathbf{U}\mathbf{K} + \lambda\mathbf{K}\mathbf{L}\mathbf{K} - \mu\mathbf{I}_m\right) \cdot \mathbf{M}. \quad (19)$$

Substituting \mathbf{Z} into (17), we obtain

$$\min_{\mathbf{M}} \quad \operatorname{tr}\left(\left(\mathbf{M}^T \mathbf{K}\mathbf{H}\mathbf{K}\mathbf{M}\right)^\dagger \mathbf{M}^T (\mathbf{K}\mathbf{U}\alpha\alpha^T \mathbf{U}\mathbf{K} + \lambda\mathbf{K}\mathbf{L}\mathbf{K} - \mu\mathbf{I}_m) \mathbf{M}\right). \quad (20)$$

Final, we obtain an equivalent maximization problem (14). \square

Being similar to TCA, the solution is finding the m leading eigenvectors of $(\mathbf{K}\mathbf{U}\alpha\alpha^T \mathbf{U}\mathbf{K} + \lambda\mathbf{K}\mathbf{L}\mathbf{K} - \mu\mathbf{I}_m)^{-1}\mathbf{K}\mathbf{H}\mathbf{K}$.

Minimizing over \mathbf{M} . The optimization over \mathbf{M} can be rewritten into the following QP form:

Input: y_s, α, X^{te} ; parameter \mathcal{P}
Output: prediction labels \mathbf{y}

- (1) Compute the weights \mathbf{w} of the classifiers.
- (2) Construct weight matrix \mathbf{W} and bias \mathbf{b} of predictors based on α .
- (3) **repeat**
- (4) Compute scores of each classifier in this category.
- (5) Find top \mathcal{P} scores.
- (6) Compute the sum of these top scores.
- (7) **until** The number of categories
- (8) Choose the max score owned category as the prediction label \mathbf{y} .

ALGORITHM 2: Ensemble Domain Adaptation Exemplar Classifiers.

$$\begin{aligned}
\min_{\alpha} \quad & \alpha^T \widehat{\mathbf{K}} \alpha - \mathbf{e}^T \alpha, \\
\text{s.t.} \quad & \alpha_0 - \sum_{i=1}^{n_s+n_T} \alpha_i = 0, \\
& 0 \leq \alpha_0 \leq C_1, \\
& 0 \leq \alpha_i \leq C_2, \\
& \forall i \geq 1.
\end{aligned} \tag{21}$$

$\widehat{\mathbf{K}} = \mathbf{UKMM}^T \mathbf{KU}$ which represents the kernel matrix has been transformed by transformation matrix \mathbf{M} . It is obvious that this problem is a QP problem and it could be solved efficiently using interior point methods or other successive optimization procedures such as Alternating Direction Method of Multipliers (ADMM).

5. Ensemble Domain Adaptation Exemplar Classifiers

In this section, we introduce the method of integration exemplar classifiers. As mentioned before, we get the number of source domain instances classifiers and this section aims to predict labels for target domain instances. In our opinions, the classification hyperplane of an exemplar classifier is representation for a source domain positive instance. However, most of the hyperplanes contain information which comes from various samples, such as images of different background or source. In fact, we aim to search the exemplar classifiers which are from instances similar to the testing sample. Thus, we utilize integrating method to filter out classifiers which include details different with the testing sample. Another view for the integration method is that it slacks the part of hyperplanes. Namely, it removes some exemplar classifiers which are trained by large instances distribution mismatch.

In our method, we first construct the classifiers from Lagrange multipliers α . The classifier construction equation is

$$\mathbf{w} = \alpha_0 \mathbf{x}^+ - \sum_{i=1}^{n_s+n_T} \alpha_i \mathbf{x}_i^-, \tag{22}$$

where \mathbf{w} is the weight of classifier.

$$b = y_j - \alpha_0 \widehat{\mathbf{K}}_{0,j} - \sum_{i=1}^{n_s+n_T} y_i \alpha_i \widehat{\mathbf{K}}_{ij}, \tag{23}$$

where b is the bias of classifier. The classifier is given by

$$s = \alpha^T \mathbf{x} + b. \tag{24}$$

And then we compute the scores by every classifier and the testing instance. Second, we find the top \mathcal{P} numbers of scores for each class classifier and compute the sum of those scores. At last, we get a score for each class, and the highest score is the category that we predict. The prediction method is described in Algorithm 2.

6. Experiments

In this section, we conduct experiments onto the four domains, Amazon, DSLR, Caltech, and Webcam, to evaluate the performance of proposed Domain Adaptation Exemplar Support Vector Machines. We first compare our method to baselines and other domain adaptation methods. Next, we analyze the effectiveness of our approach. At last, we introduce the problem of parameter sensitivity.

6.1. Data Preparation. We run the experiments on Office and Office Caltech datasets. Office dataset contains three domains Amazon, Webcam, and DSLR. Each of them includes images from amazon.com or Office environment images taken with varying lighting and pose changes using a Webcam or a DSLR camera. Office Caltech dataset contains the ten overlapping categories between the Office dataset and Caltech-256 dataset. By the standard transfer learning experiment method, we merge two datasets; it entirely includes four domains Amazon, DSLR, Caltech, and Webcam which are studied in [41]. The dataset of Amazon is the images downloaded from Amazon merchants. The images in the Webcam also come from the online web page, but they are of low quality as they are taken by web camera. The domain of DSLR is photographed by the digital SLR camera by which the images are of high quality. Caltech is always added to domain adaptation experiments, and it is collected by object detection tasks. Each domain has its characteristic. Compared to the other domains, the quality of images in

the DSLR is higher than others and the influence factors such as object detection and background are less than images downloaded from the web. Amazon and Webcam come from the web, and images in the domains are of low quality and more complexity. However, there are some different details on each of them. Instances in the Webcam are object alone, but the composition of samples in Amazon is more complex including background and other goods. Figure 1 shows the example of the backpack from four domain samples. In the view of transfer learning, the datasets come from different domains and the different margin probabilities for the images. In our model, we aim to solve this problem and get a robust classifier for the cross-domain.

We chose ten common categories among all four datasets: backpack, bike, bike helmet, bookcase, bottle, calculator, desk chair, desk lamp, desktop computer, and file cabinet. There are 8 to 151 samples per category in a domain: 958 images in Amazon, 295 images in Webcam, 157 images in DSLR, 1123 images in Caltech, and 2533 images total in the dataset. Figure 1 shows examples for datasets.

We follow both SURF and DeCAF features extraction in the experiments. First, we use SURF features encoding the images into 800-bin histograms. Next, we use DeCAF feature which is extracted by 7 layers of Alex-net [42] into 4096-bin histograms. At last, we normalized the histograms and then z -scored to have zero mean and unit standard deviation in each dimension.

We run our experiments on a standard way for visual domain adaptation. It always uses one of four datasets as source domain and another one as target domain. Each dataset provides same ten categories and uses the same representation of images which is considered as the problem of homogeneous domain adaptation. For example, we choose images taken by the set of DSLR (denoted by D) as source domain data and use images in Amazon (denoted by A) as target domain data. This problem is denoted as $\mathcal{D} \rightarrow \mathcal{A}$. Using this method, we can compose 12 domain adaptation subproblems from four domains.

6.2. Experiment Setup

(1) *Baseline Method.* We compare our DAESVM method with three kinds of classical approaches: one is classified without regularization of transfer learning, the second is conventional transfer learning methods, and the last one is the foundation model, which is low-rank exemplar support vector machine. The methods are listed as follows:

- (1) Transfer Component Analysis (TCA) [40]
- (2) Support Vector Machine (SVM) [43]
- (3) Geodesic Flow Kernel (GFK) [28]
- (4) Landmarks Selection-based Subspace Alignment (LSSA) [23]
- (5) Kernel Mean Maximum (KMM) [20]
- (6) Subspace Alignment (SA) [44]
- (7) Joint Matching Transfer (TJM) [45]
- (8) Low-Rank Exemplar-SVMs (LRESVMs) [18]

TCA, GFK, and KMM are the classical transfer learning methods. We compare our model with these methods. Besides, we prove our method is more robust than models without domain adaptation items in the transfer learning scenery. TCA is the foundation of our model, and it is similar to GFK and SFA which are based on the idea of feature transfer. KMM transfer knowledge by instance reweighting. TJM is a popular model utilizing the problem of unsupervised domain adaptation. SA and LSSA are the models using landmarks to transfer knowledge.

(2) *Implementation Details.* For baseline method, SVM is trained on the source data and tested on the target data [46]. TCA, SA, LSSA, TJM, and GFK are first viewed as dimension reduction process and then train a classifier on the source data and make a prediction for the target domain [19]. Being similar to dimension reduction, KMM is first to compute the weight of each instance and then train predictor on the reweighting source data.

Under the assumption of unsupervised domain adaptation, it is impossible to tune the optimal parameters for the target domain task by cross validation, since there exists distribution mismatch between domains. Therefore, in the experiments, we adopt the strategy of Grid Search to obtain the best parameters and report the best results. Our method involves five tunable parameters: tradeoff in ESVM C_1 and C_2 , tradeoff in regularization items λ and μ , and parameter of dimension reduction m . The parameters of tradeoff in ESVM C_1 and C_2 are selected over $\{10^{-3}, 10^{-2}, 10^{-1}, 10^0, 10^1, 10^2, 10^3\}$. We fix $\lambda = 1$, $\mu = 1$, $m = 40$ empirically and select radial basic function (RBF) as the kernel function. In fact, our model is relatively stable under a wide range of parameter values. We train a classifier for every positive instance in the source domain data and then we put them into a probability distribution. We deal with the multiclass classifier in a one versus the others way. To measure the performance of our method, we use the average accuracy and the standard deviation over ten repetitions. The average testing accuracies and standard errors for all 12 tasks of our methods are reported in Table 1. For the rest of baseline experiments, most of them are cited by the papers which are published before.

6.3. *Experiments Results.* In this section, we compare our DAESVM with baseline methods regarding classification accuracy.

Table 1 summarizes the classification accuracy obtained by all the 10 categories and generates 12 tasks in 4 domains. The highest accuracy is in a bold font which indicates that the performance of this task is better than others. First, we implement the traditional classifiers without domain adaptation items that we train the predictors on the source domain data and make a prediction for target domain dataset. Second, we compared our DAESVM with unsupervised domain adaptation methods, such as TCA or GFK, implemented to use the same dimension reduction with the parameter m in our model. At last, we also compared DAESVM with newly transfer learning models, like low-rank ESVMs [18].



FIGURE 1: Example images from the backpack category in Amazon, DSLR ((a) from left to right), Webcam, and Caltech-256 ((b) from left to right). The different domain images are various. The images have different style, background, or sources.

Overall, in a usual transfer learning way, we run datasets across different pairs of source and target domain. The accuracy of DAESVM for the adaptation from DSLR to Webcam can achieve 92.1% which make the improvement over LRESVM by 1.2%. Compared with TCA, DAESVMs make a consideration about the distribution mismatch among

instances or different domains. For the adaptation from Webcam to DSLR, this task can get the accuracy of 91.8%. For the domain datasets Amazon and Caltech which are more significant than DSLR and Webcam, DAESVM gets the accuracy of 77.5% which improves about 36.2% compared to the method of TJM. For the ability which transfers knowledge

TABLE 1: Classification accuracies of different methods for different tasks of domain adaptation. We conduct the experiments on conventional transfer learning methods. Comparing with traditional methods, DAESVMs gain a big improvement in the prediction accuracy. And they also improve confronted with the approach of LRESVM which is proposed recently [average \pm standard error of accuracy (%)].

Task	SVM	KMM	TCA	TJM	SA	GFK	LSSA	LRESVM	DAESVMs
$\mathcal{A} \rightarrow \mathcal{C}$	45.4	42.2	45.3	56.9	51.8	49.6	54.8	79.8	77.5 \pm 0.79
$\mathcal{A} \rightarrow \mathcal{D}$	50.7	42.7	60.3	56.4	56.4	55.7	57.3	74.9	76.8 \pm 0.76
$\mathcal{A} \rightarrow \mathcal{W}$	47.4	42.4	61.3	51.0	54.7	56.9	56.7	75.4	73.2 \pm 1.08
$\mathcal{C} \rightarrow \mathcal{A}$	50.7	48.3	54.7	58.6	57.1	51.2	58.4	77.2	80.2 \pm 0.39
$\mathcal{C} \rightarrow \mathcal{D}$	53.2	53.5	56.4	57.4	59.0	57.1	59.1	87.1	89.0 \pm 0.23
$\mathcal{C} \rightarrow \mathcal{W}$	44.2	45.8	50.4	58.8	62.7	57.1	58.1	74.1	74.7 \pm 0.38
$\mathcal{D} \rightarrow \mathcal{A}$	40.8	42.2	53.8	46.1	58.9	59.2	58.4	80.4	83.4 \pm 1.41
$\mathcal{D} \rightarrow \mathcal{C}$	48.3	41.6	43.9	49.6	54.3	59.4	57.7	79.0	73.0 \pm 1.04
$\mathcal{D} \rightarrow \mathcal{W}$	67.8	72.9	82.4	82.0	83.4	80.2	87.1	91.0	92.1 \pm 0.25
$\mathcal{W} \rightarrow \mathcal{A}$	42.4	41.9	53.0	50.8	57.0	66.2	59.7	74.3	77.8 \pm 0.33
$\mathcal{W} \rightarrow \mathcal{C}$	41.2	39.0	53.7	54.8	34.7	52.4	54.2	70.6	66.5 \pm 0.54
$\mathcal{W} \rightarrow \mathcal{D}$	80.2	82.0	87.9	83.4	78.9	81.2	87.2	89.2	91.8 \pm 0.59
<i>Average</i>	51.0	49.5	58.6	58.8	59.1	60.5	62.4	79.4	80.0 \pm 0.67

TABLE 2: We also conduct our experiments for the tasks of multidomain and gain an improvement comparing with methods proposed before. The experiments adopt the same strategy as the single domain adaptation. We treat multidomain as one source or target to find the shared features in a latent space. However, the complexity of the multidomain shared features limits the accuracy of tasks [average \pm standard error of accuracy (%)].

Task	SVM	KMM	TCA	TJM	SA	GFK	LSSA	LRESVM	DAESVM
$\mathcal{D}, \mathcal{W} \rightarrow \mathcal{A}$	45.7	37.4	40.5	57.1	59.4	47.3	61.7	80.1	77.2 \pm 1.27
$\mathcal{A}, \mathcal{D} \rightarrow \mathcal{C}, \mathcal{W}$	37.1	31.6	43.0	60.2	48.7	47.6	74.2	86.9	84.7 \pm 0.65
$\mathcal{D} \rightarrow \mathcal{A}, \mathcal{C}, \mathcal{W}$	41.4	43.8	57.2	63.9	51.9	51.4	77.0	82.9	88.4 \pm 0.21
$\mathcal{A}, \mathcal{D}, \mathcal{W} \rightarrow \mathcal{C}$	43.9	50.6	54.9	69.0	60.2	60.4	63.7	87.7	90.1 \pm 0.34
$\mathcal{A}, \mathcal{D} \rightarrow \mathcal{W}$	71.0	61.0	54.0	61.3	54.0	47.0	71.9	80.8	83.8 \pm 0.78
$\mathcal{A}, \mathcal{C} \rightarrow \mathcal{D}, \mathcal{W}$	81.4	53.9	77.4	71.8	57.4	64.1	80.7	89.3	92.4 \pm 0.25
<i>Average</i>	53.4	46.4	54.5	63.9	55.2	53.0	71.5	84.6	86.1 \pm 0.58

from large dataset to small domain dataset, from Amazon to DSLR, we get the accuracy of 76.8%. Contrarily, from DSLR to Amazon, the prediction accuracy is 83.4%. Totally speaking, our DAESVM trained on one domain has good performance and will also have robust performance on multidomain.

We also complement tasks of multidomains adaptation, which utilized one or more domains as source domain data and made an adaptation to other domains. The results are shown in Table 2. The accuracy of DAEVM for the adaptation from Amazon, DSLR, and Webcam to Caltech achieves 90.1% which get the improvement over LERSVM. For the task of adaptation from Amazon and Caltech to Webcam, DSLR can get the accuracy of 92.4%. The experiments prove that our models are effective not only for single domain adaptation but also for multidomain adaptation.

Two key factors may contribute to the superiority of our method: The feature transfer regularization item is utilized to slack the similarity assumption. It just assumes that there are some shared features in different domains instead of the assumption that different domains are similar to each other. This factor makes the model more robust than models with reweighting item. The second factor is the exemplar-SVMs which are proposed from a motivation of transfer learning which makes a consideration that instances are distribution

mismatch from each other. Our model combines these two factors to resist the problem of distribution mismatch among domains and sample selection bias among instances.

6.4. Pseudo Label Effectiveness. Following [19], we use pseudo labels to supplement training model. In our experiments, we test the prediction results which are influenced by the accuracy rate of pseudo labels. As a result, described by Figure 2, the prediction accuracy is improved following the increasing accuracy of pseudo labels. It is proved that the method of the pseudo label is effective and we can do the iteration by using the labels predicted by the DAESVM as the pseudo labels. The iteration step can efficiently enhance the performance of the classifiers.

6.5. Parameter Sensitivity. There are five parameters in our model, and we conduct the parameter sensitivity analysis which can achieve optimal performance under a wide range of parameter values and discuss the results.

(1) *Tradeoff λ .* λ is a tradeoff to control the weight of MMD item which aims to minimize the distribution mismatch between source and target domain. Theoretically, we want this term to be equal to zero. However, if we set this

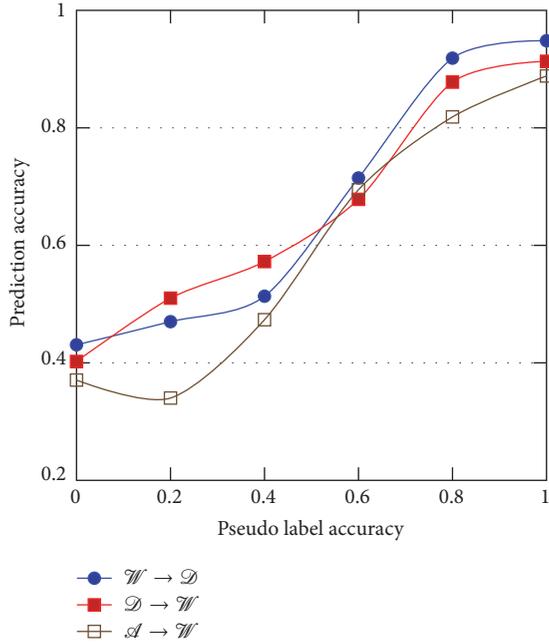


FIGURE 2: The accuracy of DAESVMs is improved with the improvement of the pseudo label accuracy. The results verify the effectiveness of the pseudo label method.

parameter to infinite, $\lambda \rightarrow \infty$, it may lose the data properties when we transform source and target domain data into high-dimension space. Contrarily, if we set λ to zero, the model would lose the function of correcting the distribution mismatch.

(2) *Tradeoff μ* . μ is a tradeoff to control the weight of data variance item which aims to preserve data properties. Theoretically, we want this item to be equal to zero. However, if we set this parameter to infinite, $\mu \rightarrow \infty$, it may augment the data distribution mismatch among different domains; namely, transformation matrix \mathbf{M} cannot utilize source data to assist the target task. Contrarily, if we set μ to zero, the model cannot preserve the properties of original data.

(3) *Dimension Reduction m* . m is the dimension of the transformation matrix, namely, the dimension of the subspace which we want to map samples into. Similarly, minimizing m too less may lead to losing the properties of data which may lead to the classifier failure. If m is too large, the effectiveness of correct distribution mismatch may be lost. We conduct the classification results influenced by the dimension of m , and the results are displayed in Figure 3.

(4) *Tradeoff in ESVM C_1 and C_2* . Parameters C_1 and C_2 are the upper bound of the Lagrangian variables. In the standard SVM, positive and negative instances share the same standard of these two parameters. In our models, we expect the weights of the positive samples to be higher than negative samples. In our experiments, the value of C_1 is one hundred times C_2 which could gain a high-performance predictor. The visual analysis of these two parameters is in Figure 4.

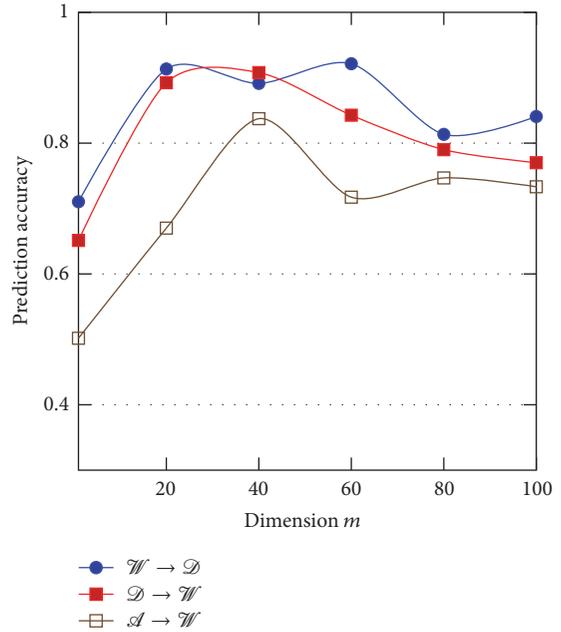


FIGURE 3: When the dimension is 20 or 40, the prediction accuracy is higher than others.

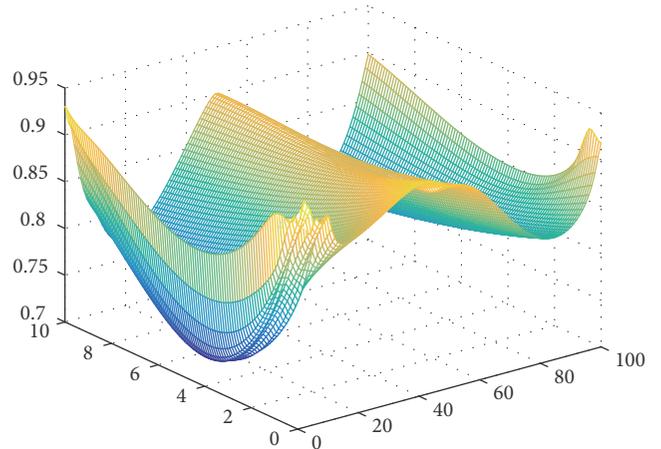


FIGURE 4: We fix $\lambda = 1$, $m = 20$, and $\mu = 1$ in these experiments, and C_1 is searched in $\{0.1, 0.5, 1, 5, 10, 50, 100\}$ and C_2 is searched in $\{0.001, 0.005, 0.01, 0.1, 0.5, 1, 10\}$.

7. Conclusion

In this paper, we have proposed an effective method for domain adaptation problems with regularization item which reduces the data distribution mismatch between domains and preserves properties of the original data. Furthermore, utilizing the method of integrating classifiers can predict target domain data with high accuracy. The proposed method mainly aims to solve the problem, in which domains or instances distributions mismatch occurs. Meanwhile, we extend DAESVMs to the multiple source or target domains. Experiments conducted on the transfer learning datasets transfer knowledge from image to image.

Our future works are as follows. First, we will integrate the training procession of all the classifiers in an ensemble way. It is better to accelerate training process by rewriting all the weight into a matrix form. This strategy can omit the process of matrix inversion optimization. Second, we want to make a constraint for α that can hold the sparsity. At last, we will extend DAESVMs on the problem transfer knowledge among domains which have few relationships, such as transfer knowledge from image to video or text.

Notations and Descriptions

$\mathcal{D}_S, \mathcal{D}_T$:	Source/target domain
$\mathcal{T}_S, \mathcal{T}_T$:	Source/target task
d :	Dimension of feature
$\mathbf{X}_S, \mathbf{X}_T$:	Source/target sample matrix
$\mathbf{y}_S, \mathbf{y}_T$:	Source/target sample label matrix
\mathbf{K} :	Kernel matrix without label information
α :	Lagrange multipliers vector
n_S, n_T :	The number of source/target domain instances
\mathbf{e} :	Identity vector
\mathbf{I} :	Identity matrix.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work has been partially supported by grants from National Natural Science Foundation of China (nos. 61472390, 71731009, 91546201, and 11771038) and the Beijing Natural Science Foundation (no. 1162005).

References

- [1] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1137–1149, 2017.
- [2] R. Collobert and J. Weston, "A unified architecture for natural language processing: deep neural networks with multitask learning," in *Proceedings of the 25th International Conference on Machine Learning*, pp. 160–167, ACM, July 2008.
- [3] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the 27th IEEE Conference on Computer Vision and Pattern Recognition (CVPR '14)*, pp. 580–587, Columbus, Ohio, USA, June 2014.
- [4] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 10, pp. 1345–1359, 2010.
- [5] W.-S. Chu, F. D. L. Torre, and J. F. Cohn, "Selective transfer machine for personalized facial action unit detection," in *Proceedings of the 26th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2013*, pp. 3515–3522, USA, June 2013.
- [6] A. Kumar, A. Saha, and H. Daume, "Co-regularization based semi-supervised domain adaptation," in *Advances in Neural Information Processing Systems 23*, pp. 478–486, 2010.
- [7] M. Xiao and Y. Guo, "Feature space independent semi-supervised domain adaptation via kernel matching," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 1, pp. 54–66, 2015.
- [8] S. J. Pan, J. T. Kwok, Q. Yang, and J. J. Pan, "Adaptive localization in a dynamic WiFi environment through multi-view learning," in *Proceedings of the AAAI-07/IAAI-07 Proceedings: 22nd AAAI Conference on Artificial Intelligence and the 19th Innovative Applications of Artificial Intelligence Conference*, pp. 1108–1113, can, July 2007.
- [9] A. Van Engelen, A. C. Van Dijk, M. T. B. Truijman et al., "Multi-Center MRI Carotid Plaque Component Segmentation Using Feature Normalization and Transfer Learning," *IEEE Transactions on Medical Imaging*, vol. 34, no. 6, pp. 1294–1305, 2015.
- [10] Y. Zhang, J. Wu, Z. Cai, P. Zhang, and L. Chen, "Memetic Extreme Learning Machine," *Pattern Recognition*, vol. 58, pp. 135–148, 2016.
- [11] M. Uzair and A. Mian, "Blind domain adaptation with augmented extreme learning machine features," *IEEE Transactions on Cybernetics*, vol. 47, no. 3, pp. 651–660, 2017.
- [12] L. Zhang and D. Zhang, "Domain Adaptation Extreme Learning Machines for Drift Compensation in E-Nose Systems," *IEEE Transactions on Instrumentation and Measurement*, vol. 64, no. 7, pp. 1790–1801, 2015.
- [13] B. Scholkopf, J. Platt, and T. Hofmann, in *A kernel method for the two-sample-problem*, pp. 513–520, 2008.
- [14] J. Wu, S. Pan, X. Zhu, C. Zhang, and X. Wu, "Multi-instance Learning with Discriminative Bag Mapping," *IEEE Transactions on Knowledge and Data Engineering*, pp. 1-1.
- [15] J. Wu, S. Pan, X. Zhu, C. Zhang, and P. S. Yu, "Multiple Structure-View Learning for Graph Classification," *IEEE Transactions on Neural Networks and Learning Systems*, vol. PP, no. 99, pp. 1–16, 2017.
- [16] T. Malisiewicz, A. Gupta, and A. A. Efros, "Ensemble of exemplar-SVMs for object detection and beyond," in *Proceedings of the 2011 IEEE International Conference on Computer Vision, ICCV 2011*, pp. 89–96, Spain, November 2011.
- [17] B. Zadrozny, "Learning and evaluating classifiers under sample selection bias," in *Proceedings of the Twenty-first international conference*, p. 114, Banff, Alberta, Canada, July 2004.
- [18] W. Li, Z. Xu, D. Xu, D. Dai, and L. Van Gool, "Domain Generalization and Adaptation using Low Rank Exemplar SVMs," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1-1.
- [19] M. Long, J. Wang, G. Ding, S. J. Pan, and P. S. Yu, "Adaptation regularization: A general framework for transfer learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, no. 5, pp. 1076–1089, 2014.
- [20] J. Huang, A. J. Smola, A. Gretton, K. M. Borgwardt, and B. Schölkopf, "Correcting sample selection bias by unlabeled data," in *Proceedings of the 20th Annual Conference on Neural Information Processing Systems, NIPS 2006*, pp. 601–608, can, December 2006.
- [21] M. Sugiyama and M. Kawanabe, *Machine Learning in Non-Stationary Environments*, The MIT Press, 2012.
- [22] W. Dai, Q. Yang, G. Xue, and Y. Yu, "Boosting for transfer learning," in *Proceedings of the 24th International Conference on*

- Machine Learning (ICML '07)*, pp. 193–200, New York, NY, USA, June 2007.
- [23] R. Aljundi, R. Emonet, D. Muselet, and M. Sebban, “Landmarks-based kernelized subspace alignment for unsupervised domain adaptation,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015*, pp. 56–63, USA, June 2015.
- [24] B. Tan, Y. Zhang, S. J. Pan, and Q. Yang, “Distant domain transfer learning,” in *Proceedings of the 31st AAAI Conference on Artificial Intelligence, AAAI 2017*, pp. 2604–2610, usa, February 2017.
- [25] W.-S. Chu, F. De La Torre, and J. F. Cohn, “Selective transfer machine for personalized facial expression analysis,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 3, pp. 529–545, 2017.
- [26] R. Aljundi, J. Lehaire, F. Prost-Boucle, O. Rouvière, and C. Lartizien, “Transfer learning for prostate cancer mapping based on multicentric MR imaging databases,” *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics): Preface*, vol. 9487, pp. 74–82, 2015.
- [27] M. Long, *Transfer learning: problems and methods [Ph.D. thesis]*, Tsinghua University, problems and methods. PhD thesis, 2014.
- [28] B. Gong, Y. Shi, F. Sha, and K. Grauman, “Geodesic flow kernel for unsupervised domain adaptation,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '12)*, pp. 2066–2073, June 2012.
- [29] J. Blitzer, R. McDonald, and F. Pereira, “Domain adaptation with structural correspondence learning,” in *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP '06)*, pp. 120–128, Association for Computational Linguistics, July 2006.
- [30] M. Long, Y. Cao, J. Wang, and M. I. Jordan, “Learning transferable features with deep adaptation networks,” and M. I. Jordan. *Learning transferable features with deep adaptation networks. pages 97–105*, pp. 97–105, 2015.
- [31] M. Long, J. Wang, and M. I. Jordan, *Deep transfer learning with joint adaptation networks*, 2016.
- [32] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, “How transferable are features in deep neural networks?” in *Proceedings of the 28th Annual Conference on Neural Information Processing Systems 2014, NIPS 2014*, pp. 3320–3328, can, December 2014.
- [33] J. Yang, R. Yan, and A. G. Hauptmann, “Cross-domain video concept detection using adaptive SVMs,” in *Proceedings of the 15th ACM International Conference on Multimedia (MM '07)*, pp. 188–197, September 2007.
- [34] S. Li, S. Song, and G. Huang, “Prediction reweighting for domain adaption,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 28, no. 7, pp. 1682–1695, 2017.
- [35] Z. Xu, W. Li, L. Niu, and D. Xu, “Exploiting low-rank structure from latent domains for domain generalization,” *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics): Preface*, vol. 8691, no. 3, pp. 628–643, 2014.
- [36] L. Niu, W. Li, D. Xu, and J. Cai, “An Exemplar-Based Multi-View Domain Generalization Framework for Visual Recognition,” *IEEE Transactions on Neural Networks and Learning Systems*, 2016.
- [37] L. Niu, W. Li, and D. Xu, “Multi-view domain generalization for visual recognition,” in *Proceedings of the 15th IEEE International Conference on Computer Vision, ICCV 2015*, pp. 4193–4201, Chile, December 2015.
- [38] T. Kobayashi, “Three viewpoints toward exemplar SVM,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015*, pp. 2765–2773, USA, June 2015.
- [39] S. J. Pan, J. T. Kwok, and Q. Yang, “Transfer learning via dimensionality reduction,” in *In Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 677–682, 2008.
- [40] S. J. Pan, I. W. Tsang, J. T. Kwok, and Q. Yang, “Domain adaptation via transfer component analysis,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 22, no. 2, pp. 199–210, 2011.
- [41] K. Saenko, B. Kulis, M. Fritz, and T. Darrell, “Adapting visual category models to new domains,” in *Computer Vision—ECCV 2010*, vol. 6314 of *Lecture Notes in Computer Science*, pp. 213–226, Springer, Berlin, Germany, 2010.
- [42] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” in *Proceedings of the 26th Annual Conference on Neural Information Processing Systems (NIPS '12)*, pp. 1097–1105, Lake Tahoe, Nev, USA, December 2012.
- [43] V. N. Vapnik, *Statistical Learning Theory*, Adaptive and Learning Systems for Signal Processing, Communications, and Control, Wiley- Interscience, New York, NY, USA, 1998.
- [44] B. Fernando, A. Habrard, M. Sebban, and T. Tuytelaars, “Unsupervised visual domain adaptation using subspace alignment,” in *Proceedings of the 2013 14th IEEE International Conference on Computer Vision, ICCV 2013*, pp. 2960–2967, Australia, December 2013.
- [45] M. Long, J. Wang, G. Ding, J. Sun, and P. S. Yu, “Transfer joint matching for unsupervised domain adaptation,” in *Proceedings of the 27th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2014*, pp. 1410–1417, USA, June 2014.
- [46] C. Chang and C. Lin, “LIBSVM: a Library for support vector machines,” *ACM Transactions on Intelligent Systems and Technology*, vol. 2, no. 3, article 27, 2011.

Research Article

Exploiting Interslice Correlation for MRI Prostate Image Segmentation, from Recursive Neural Networks Aspect

Qikui Zhu,¹ Bo Du ,¹ Baris Turkbey,² Peter Choyke,² and Pingkun Yan³

¹School of Computer, Wuhan University, Wuhan, China

²National Cancer Institute, National Institutes of Health, Bethesda, MD, USA

³Biomedical Engineering Department, Rensselaer Polytechnic Institute, Troy, NY 12180, USA

Correspondence should be addressed to Bo Du; remoteking@whu.edu.cn

Received 27 August 2017; Accepted 8 January 2018; Published 28 February 2018

Academic Editor: Gang Li

Copyright © 2018 Qikui Zhu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Segmentation of the prostate from Magnetic Resonance Imaging (MRI) plays an important role in prostate cancer diagnosis. However, the lack of clear boundary and significant variation of prostate shapes and appearances make the automatic segmentation very challenging. In the past several years, approaches based on deep learning technology have made significant progress on prostate segmentation. However, those approaches mainly paid attention to features and contexts within each single slice of a 3D volume. As a result, this kind of approaches faces many difficulties when segmenting the base and apex of the prostate due to the limited slice boundary information. To tackle this problem, in this paper, we propose a deep neural network with bidirectional convolutional recurrent layers for MRI prostate image segmentation. In addition to utilizing the intraslice contexts and features, the proposed model also treats prostate slices as a data sequence and utilizes the interslice contexts to assist segmentation. The experimental results show that the proposed approach achieved significant segmentation improvement compared to other reported methods.

1. Introduction

Accurately segmenting the prostate from Magnetic Resonance Imaging (MRI) provides very useful information for clinical applications like computer aided diagnosis and image guided interventions [1]. However, it is a very challenging task due to the lack of clear boundary definition and the significant variation of shape and texture across images from different patients [2] as shown in Figure 1.

According to the guidance information used in the model, we can classify the existing prostate MRI segmentation methods into four kinds: region based, shape prior based, contour based, and classification methods [3–5]. Toth et al. [6] presented an Active Shape Model (ASM) initialization scheme for prostate segmentation which leverages multimodal information to initialize ASM. Samiee et al. [7] proposed a model using shape prior of prostate to refine the prostate boundary; Klein [8] proposed an automatic segmentation model which is based on manually matching segmented atlas images.

In the past several years, deep learning based techniques, especially fully convolutional neural networks (FCNs), have proved very effective on image segmentation [9–11], including biomedical image segmentation. Zhu et al. [12] proposed a deeply supervised CNN that utilized the residual information to accurately segment the prostate MRI. Bao and Chung [13] introduced a multiscale structured FCN model for brain MRI segmentation by capturing discriminative features from input patch. Other examples on introducing deep learning into biomedical image segmentation can be found in [14–17].

However, a straightforward extension of those 2D image segmentation methods to 3D may not yield satisfactory performance, due to the anisotropic nature of many medical imaging modalities. To tackle this problem, Chen et al. [18] proposed a method to combine fully convolutional neural networks with the extended Convolutional Long Short-Term Memory (C-LSTM), which improved 3D medical image segmentation performances by simultaneously leveraging the abstraction capabilities of both FCNs and RNNs. Nevertheless, their model relies on the so-called U-Net [18] to extract

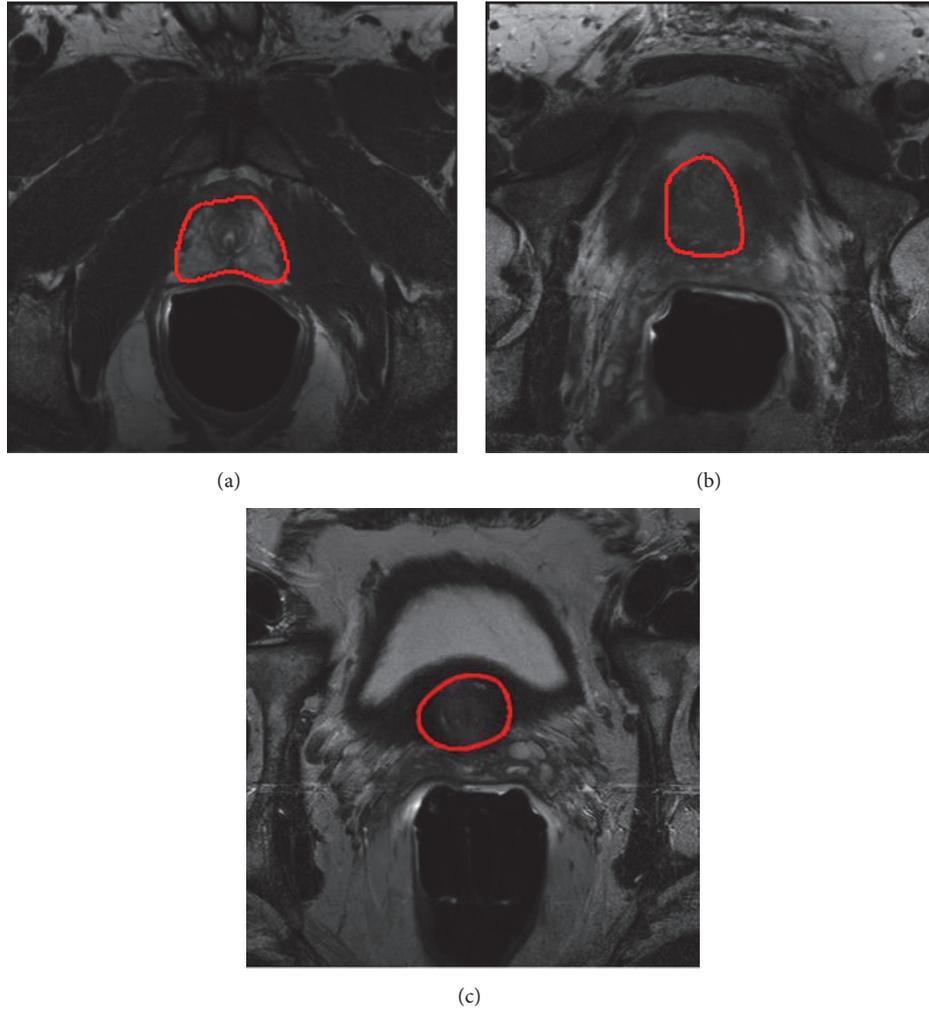


FIGURE 1: Challenges in segmenting prostate in MR images. (a) Imaging artifacts inside the bladder. (b) Weak boundary. (c) Complex intensity distribution.

image features. The following Bidirectional Convolutional LSTMs (BDC-LSTMs) only work on the extracted features. Thus, useful information for assisting image segmentation may be lost.

As to prostate segmentation task, an insurmountable challenge is the image slice specifically at the apex and base always loses clear boundaries and necessary information. This phenomenon brings the main difficulty to prostate segmentation. However, shape prior is an effective way to resolve this challenge. For instance, Qin et al. [19] proposed an adaptive shape prior constrained directional level set model (ASPDLS) to segment the inner and outer boundaries of the bladder wall and achieved accurate segmentation results. Motivated by the fact that the acquired MRI images typically have a high intraslice resolution and there exists a high spatial dependence between slices from the same patient, we utilize interslice as shape prior to guide the process of feature extraction and explore necessary information from interslice to alleviate information loss as shown in Figure 2. Besides the architecture of RNNs has superiority performances in modeling sequential data [17, 20, 21]. To improve the performance of prostate segmentation, in this paper, we propose



FIGURE 2: Interslice as shape prior to guide the process of feature extraction.

a network, called UR-Net, which treats prostate slices as a data sequence, utilizing the intraslice contexts and features to assist segmentation.

There are two main contributions of this paper. First, we treat prostate slices as a data sequence and utilize interslice as shape prior to guide the process of feature extraction and explore necessary information from interslice. Second, we explore the power of RNNs rather than the traditional CNNs to extract image feature. The experimental results demonstrate that the use of RNNs can substantially improve the performance of prostate segmentation.

The rest of the paper is organized as follows. The architecture of Recurrent Neural Network and the details of proposed network architecture are described in Section 2.

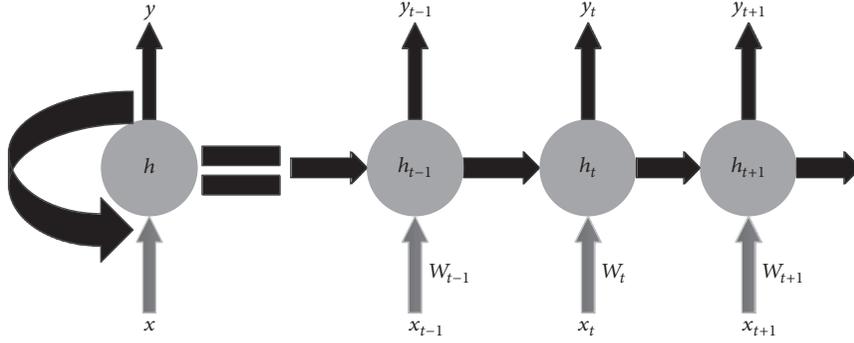


FIGURE 3: Architecture of the Recurrent Neural Networks and unfolding Recurrent Neural Networks into a full network. x_t is the input at time step t . h_t is the hidden state at time step t , which is the “memory” of the network. y_t is the output at time step t .

Section 3 presents the experimental results and performance evaluation. The conclusions are provided in Section 4.

2. UR-Net

In this section, we first review the classic Recurrent Neural Networks (RNNs) and then move on to describe the extension of RNNs to Long Short-Term Memory (LSTM) [22] and Convolutional Long Short-Term Memory (CLSTM) [18] which are specific Recurrent Neural Networks. After that, the proposed UR-Net is presented in detail.

2.1. Recurrent Neural Networks (RNNs). Recurrent Neural Network (RNN) has a long history in the artificial neural network community which was designed to model temporal sequences. The architecture of typical RNNs is shown in Figure 3. This model has shown great promise in many tasks, such as NLP [23], non-Markovian control, and text tasks [24]. The idea behind RNNs is to make use of sequential information with the output being dependent on the previous computation. RNNs have a memory, which can remember the information about what has been calculated so far. In theory [25], RNNs can remember the information in arbitrarily long sequences and make use of the previous computations, but in practice they are limited to looking back only a few steps, because of the problem of vanishing gradient.

At each time step t , the RNNs utilize the input data x_t and the previous hidden state h_{t-1} to calculate the next hidden state h_t and output y_t by applying the following recursive operation:

$$\begin{aligned} h_t &= f_h(W_h x_t + U h_{t-1} + b_h), \\ y_t &= f_o(W_o h_t + b_o), \end{aligned} \quad (1)$$

where f is an element-wise nonlinearity function; W_h , U , and b_h are the parameters of hidden state; W_o and b_o are output parameters.

During the last decade, several methods have been explored for training RNNs, such as backpropagation through time (BPTT) [26], real-time recurrent learning (RTRL) [27], and extended Kalman filtering based techniques (EKF) [28]. Though those training methods can help us train RNNs, they suffer from the vanishing gradient problem.

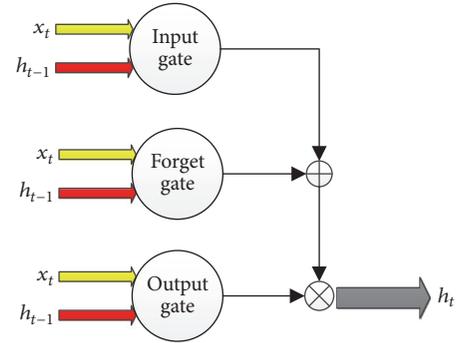


FIGURE 4: Architecture of the Long Short-Term Memory.

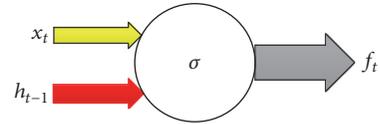


FIGURE 5: Architecture of the forget gate in LSTM.

2.2. Long Short-Term Memory (LSTM). To address the problem of vanishing gradient and long-term dependency residing in RNNs [29], a special implementation of RNN, Long Short-Term Memory (LSTM), was introduced by Hochreiter and Schmidhuber [22]. The architecture of LSTM is shown in Figure 4.

One LSTM unit consists of an input gate (i), a forget gate (f), an output gate (o), and a memory cell (c) which possess the ability of remembering or forgetting the information over potentially long periods of time. The input gate puts and controls the input data into the memory cell. The forgetting gate decides what information we are going to throw away from the memory cell. The output gate decides which parts of data in the memory cell are going to output and simultaneously controls the output data flow into the rest of the network.

The step of LSTM can be described as follows: (1) the first step in the LSTM model corresponds to (2) which is to decide what information should be discarded from input data. This decision is made by forget gate; the architecture of forget gate is shown in Figure 5. The forget gate possesses a forgetting

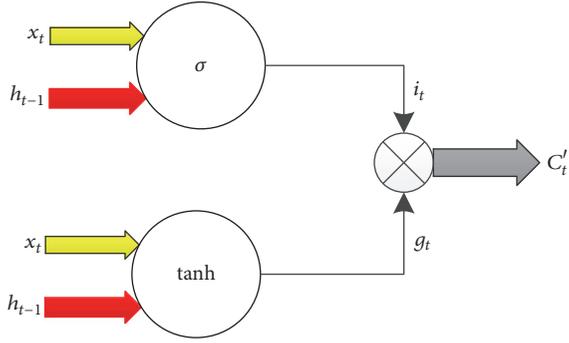


FIGURE 6: Architecture of the input gate in LSTM.

layer which consists of a sigmoid function. When getting the data from previous stage h_{t-1} and input data x_t , the sigmoid function outputs a number 0 or 1 for each data in the cell state C_{t-1} , the number 0 represents completely throwing it away, while 1 represents storing it.

$$f_t = \sigma(W^f x_t + U^f h_{t-1} + b^f). \quad (2)$$

(2) The second step is to decide what new information we should store in the cell. The input gate consists of a sigmoid function and a tanh function as shown in Figure 6. When the input gate receives a new data, the sigmoid function decides what values will be updated and the tanh function creates a candidate values g_t , those operations corresponding to (3). At last, the input gate controls the candidate values to update the cell state.

$$\begin{aligned} i_t &= \sigma(W^i x_t + U^i h_{t-1} + b^i), \\ g_t &= \tanh(W^g x_t + U^g h_{t-1} + b^g). \end{aligned} \quad (3)$$

(3) When we got the information coming from forget gate and input gate, we can utilize (4) to update the cell state. This operation can drop some useless information.

$$c_t = f_t \circ c_{t-1} + i_t \circ g_t. \quad (4)$$

(4) The fourth step is to decide what information we should output which is based on cell state. The final result consists of two parts, the first part comes from cell state selected by a tanh function. The second part comes from input data which will be selected by a sigmoid function. The output gate is shown in Figure 7 and the computing methods are

$$\begin{aligned} o_t &= \sigma(W^o x_t + U^o h_{t-1} + b^o), \\ h_t &= o_t \circ \tanh(c_t). \end{aligned} \quad (5)$$

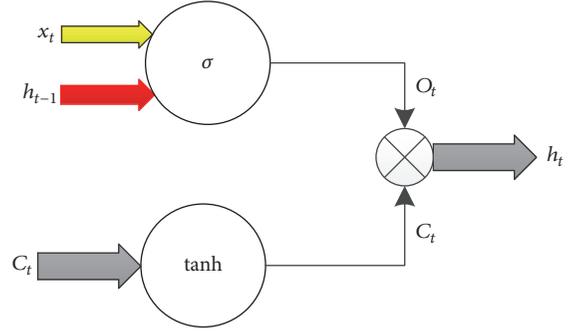


FIGURE 7: Architecture of the output gate in LSTM.

Putting those together, the gates at discrete time t ($t = 1, 2, \dots$) are computed as follows:

$$\begin{aligned} i_t &= \sigma(W^i x_t + U^i h_{t-1} + b^i), \\ f_t &= \sigma(W^f x_t + U^f h_{t-1} + b^f), \\ o_t &= \sigma(W^o x_t + U^o h_{t-1} + b^o), \\ g_t &= \tanh(W^g x_t + U^g h_{t-1} + b^g), \\ c_t &= f_t \circ c_{t-1} + i_t \circ g_t, \\ h_t &= o_t \circ \tanh(c_t). \end{aligned} \quad (6)$$

The standard LSTM architecture is just designed for one-dimensional data. It cannot be directly applied to 2D image data. To apply LSTM in prostate image, in our model, we apply Convolutional LSTM (CLSTM) as convolutional layer. This can be achieved by using a convolution operator to replace the matrix multiplication. The core equations of CLSTM are presented in

$$\begin{aligned} i_t^d &= \sigma(W_i^d * x_t^d + U_i^d h_{t-1}^d + b_i^d), \\ f_t^d &= \sigma(W_f^d * x_t^d + U_f^d h_{t-1}^d + b_f^d), \\ o_t^d &= \sigma(W_o^d * x_t^d + U_o^d h_{t-1}^d + b_o^d), \\ g_t^d &= \tanh(W_g^d * x_t^d + U_g^d h_{t-1}^d + b_g^d), \\ c_t^d &= f_t^d \circ c_{t-1}^d + i_t^d \circ g_t^d, \\ h_t^d &= o_t^d \circ \tanh(c_t^d), \end{aligned} \quad (7)$$

where $(*)$ denotes convolution operator and h is the output of the layer; d denotes the fact that the CLSTM works slice by slice in certain direction.

2.3. The Proposed Network Architecture. In order to exploit the interslice information effectively, we introduce a Bidirectional Convolutional LSTM (BDC-LSTM) layer into our deep learning network. A BDC-LSTM layer consists of two sets of CLSTMs to extract features as shown in Figure 8. The two CLSTM streams work in two opposite directions. Rather than

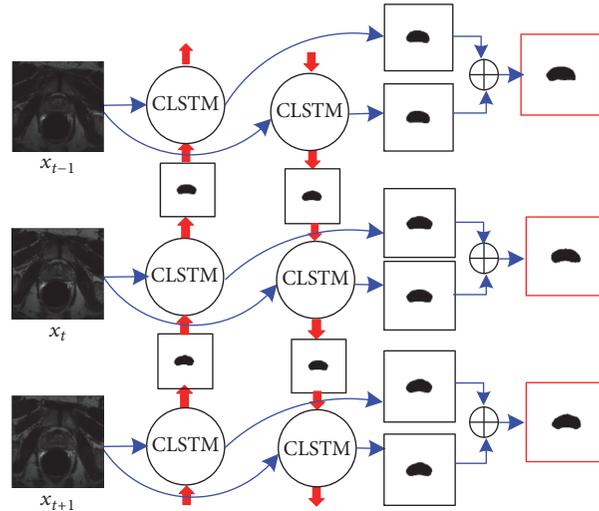


FIGURE 8: The architecture of BDC-LSTM. One BDC-LSTM layer contains two CLSTM layers working in two opposite directions. One from x_{t+1} to x_{t-1} and the other from x_{t-1} to x_{t+1} . The results of two CLSTM layers are connected together as the input of next layer.

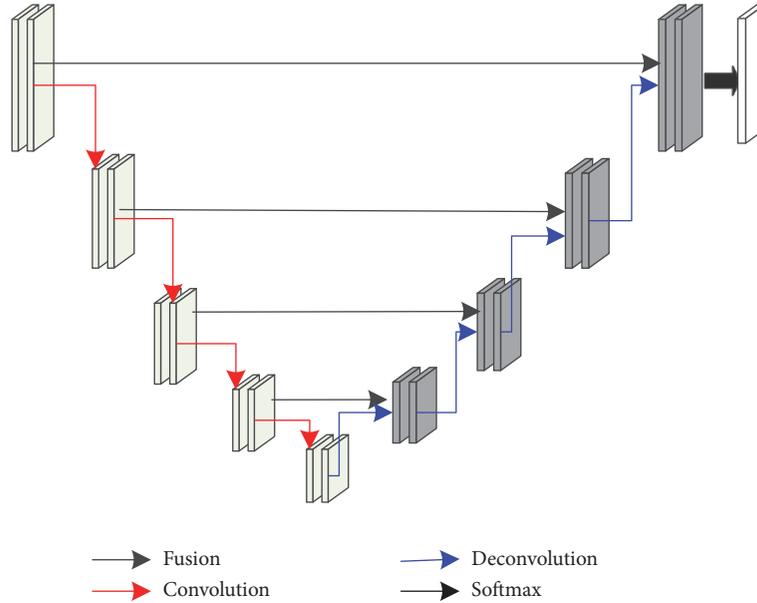


FIGURE 9: Architecture of U-Net.

serializing each prostate image into sequential patches and then leveraging Bidirectional LSTM to segment each patch, in our method, we treat each image as a whole and three adjacent image slices compose a sequence.

When we put one image sequence denoted by x_{t-1} , x_t , x_{t+1} into the BDC-LSTM layer as shown in Figure 8, the layer will utilize interslice and intraslice information to extract prostate features. Firstly, the layer extracts the first slice's x_{t-1} features. And then the result of x_{t-1} and x_t will be treated as a shape prior combined with the later slice x_t and x_{t+1} as input to guide the process of segmentation in turn. Simultaneously the layer will extract features of each slice in opposite directions from x_{t+1} to x_{t-1} , at last, connecting the two different features maps together as the input of next layer.

Our proposed network architecture is shown in Figure 10. The main framework of our proposed method follows the architecture of U-Net [30], since U-Net can successfully extract image features for segmentation with a reasonable network depth. As a matter of fact, U-Net has obtained state-of-the-art performances in many biomedical image processing tasks. For example, Milletari et al. [31] proposed a fully convolutional neural network for volumetric medical image segmentation, called V-Net. This model leverages the power of U-Net to process MRI volumes. The architecture of U-Net is shown in Figure 9.

The proposed network architecture consists of a contracting path on the left, an expansive path on the right, and a classified path on the bottom. Both the contracting path and

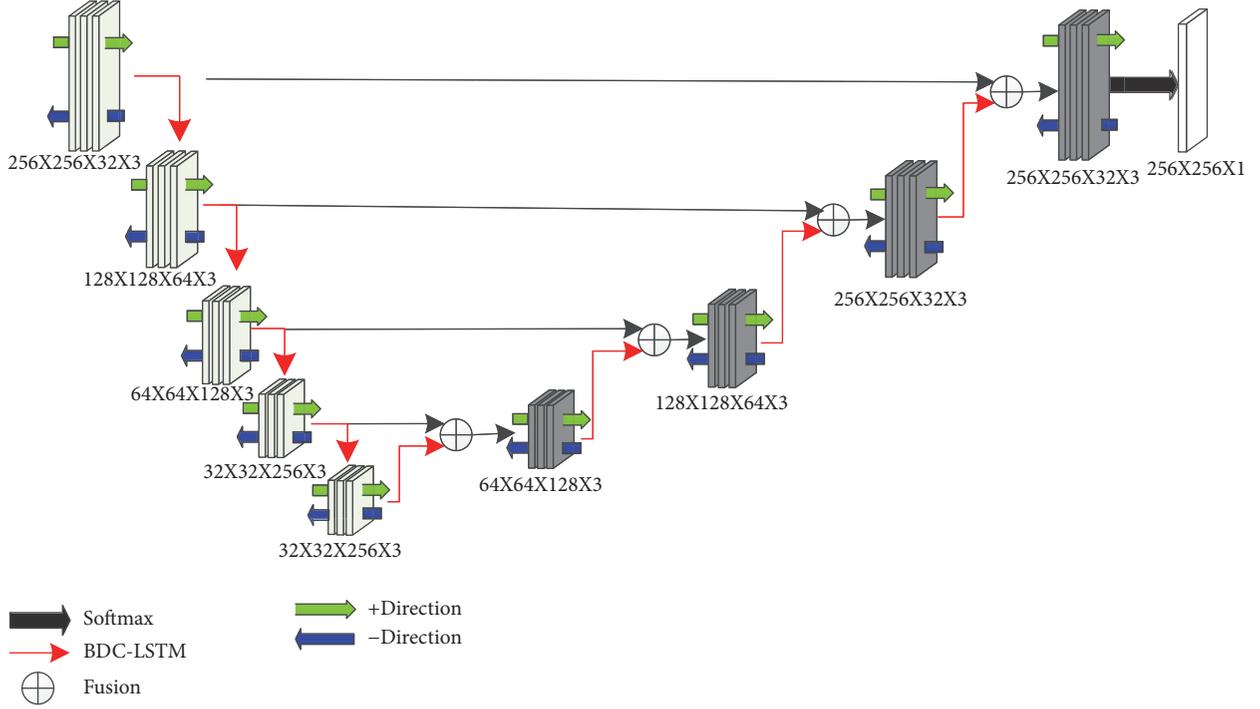


FIGURE 10: Architecture of the proposed network.

expansive path possess 4 stages, and each stage consists of one BDC-LSTM layer. At the end of networks one softmax layer is added. At the contracting stage, a 2×2 max pooling operation with a stride of 2 is attached for downsampling and the number of feature channels is doubled after each stage. On the opposite side, at the expansive stage, the first step is upsampling, which makes the width and height of feature maps doubled each time until they reach the size of the original images. At the same time, upsampling also halves the number of feature channels. To reduce information loss during convolution, a concatenation from left contracting path to right expansive path is made. The concatenation can provide features extracted from early stage to late stage and also can speed up the convergence of the network. To avoid overfitting, dropout operations have been added at the end of each stage.

2.4. Network Objective Function. For the prostate images, the anatomy of interest usually occupies a very small part of an image. This brings the problem that the networks always ignore the segmentation parts and become biased towards the background. This always led to the learning process trapped in local minima. To overcome this problem, we apply the dice coefficient as the objective function. The dice coefficient function can pay more attention to segmentation parts influences. The dice coefficient (DSC) [19] function between two images can be written as

$$\text{DSC}(S_a, S_m) = \frac{2|S_a \cap S_m|}{|S_a| + |S_m|}, \quad (8)$$

where S_a denotes the result of automatic segmentation and S_m denotes the result of manual segmentation.

In our work, the ground truth and results of segmentation are binary images, so the dice coefficient DSC between two binary images can be written as

$$\text{DSC} = \frac{2 \sum_i^N p_i q_i}{\sum_i^N p_i^2 + \sum_i^N q_i^2}, \quad (9)$$

N denotes the total number of pixels in the image, and p_i , q_i denote the pixels from ground truth and segmentation, respectively.

This formulation of dice can be differentiated yielding the gradient:

$$\frac{\partial D}{\partial p_j} = 2 \left[\frac{q_j (\sum_i^N p_i^2 + \sum_i^N q_i^2) - 2p_j \sum_i^N p_i q_i}{(\sum_i^N p_i^2 + \sum_i^N q_i^2)^2} \right]. \quad (10)$$

Besides, Milletari et al. [31] have proved that the DSC are much better than the same network trained with a logistic loss for overcoming the network traps in local minima.

3. Experimental Results

3.1. Materials. The MRI prostate images used in our work as shown in Figure 11 were acquired from 80 patients using a Philips 3T MRI scanner with endorectal coil. The in-plane resolution is $0.3 \text{ mm} \times 0.3 \text{ mm}$ and interslice distance is 3 mm. Each patient image volume consists of about 26 slices. The dimension of each 2D slice is 512×512 pixels.

3.2. Training Strategy. We randomly selected 76 patients from 80 patients for training and the rest of patients are utilized

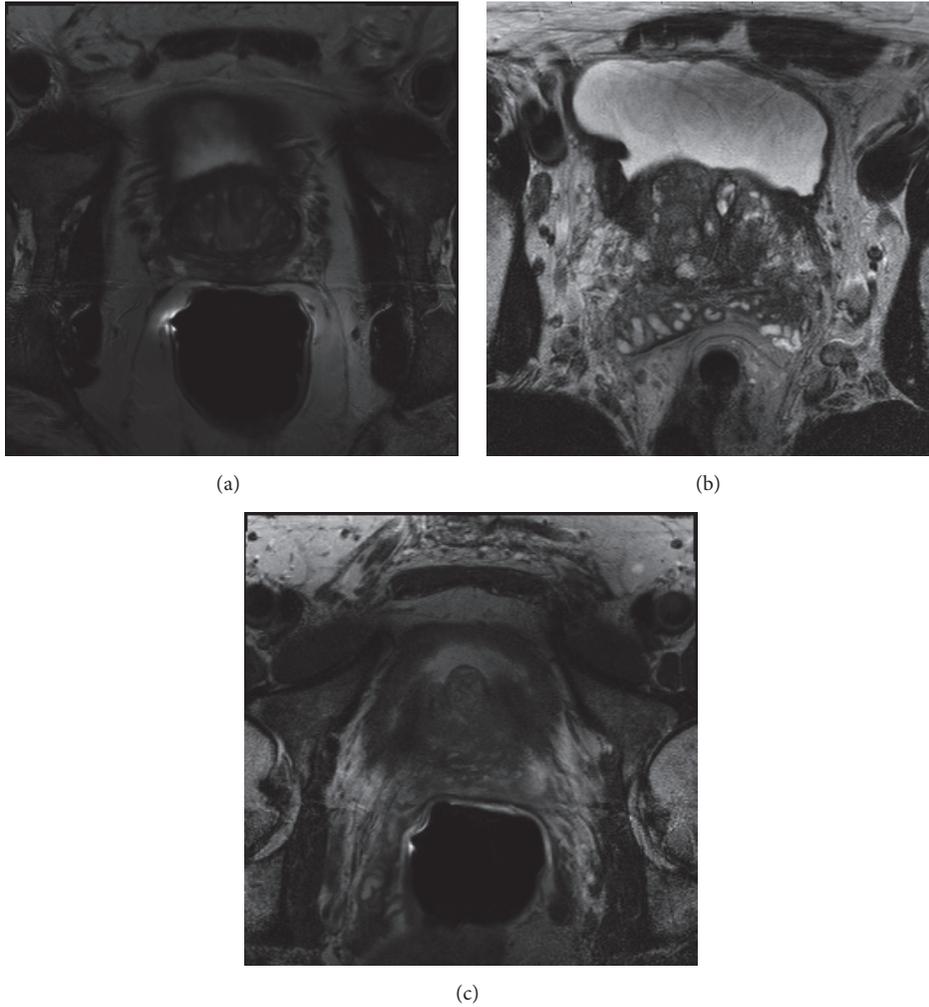
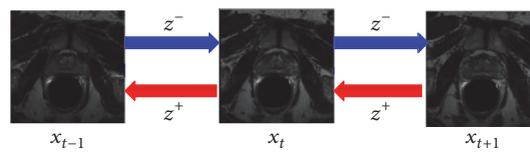


FIGURE 11: Parts of MRI prostate images.

FIGURE 12: The z^- direction and z^+ direction.

for testing. During training, we put three sequential slices denoted by x_{t-1} , x_t , x_{t+1} from one patient into the network. And then the BDC-LSTM layers exploit intraslice and interslice contextual information from two directions, one in z^- direction and the other in z^+ direction as shown in Figure 12. Our network is trained end-to-end on the prostate scans dataset. And the network framework is implemented under the open-source deep learning library Keras [32]. Experiments are carried out on GTX1080 GPU with 8 GB of video memory and the CUDA edition is 8.0. In the training phase, the learning rate is set as 0.0001 initially. Due to the limit by the memory, we choose 1 as the mini-batch. And all of the train image and ground truth have been resized to 256×256 .

3.3. Experiments. To validate whether the deep neural network with RNN layers can significantly improve the segmentation accuracy, we also modify the FCNs by utilizing BDC-LSTM layers to replace the convolutional layers within FCNs. These testing images come with a corresponding ground truth segmentation map which is a binary image and is used to evaluate the performances of automatic segmentation. At last, we compare our model with U-Net, V-Net [31], fully convolutional networks (FCNs), and modified FCNs. Parts of segmentation results of our network are shown in Figure 13.

3.3.1. Qualitative Comparison. From the segmentation results, we selected some representative and challenging images, which have fuzzy boundaries and the pixel intensity

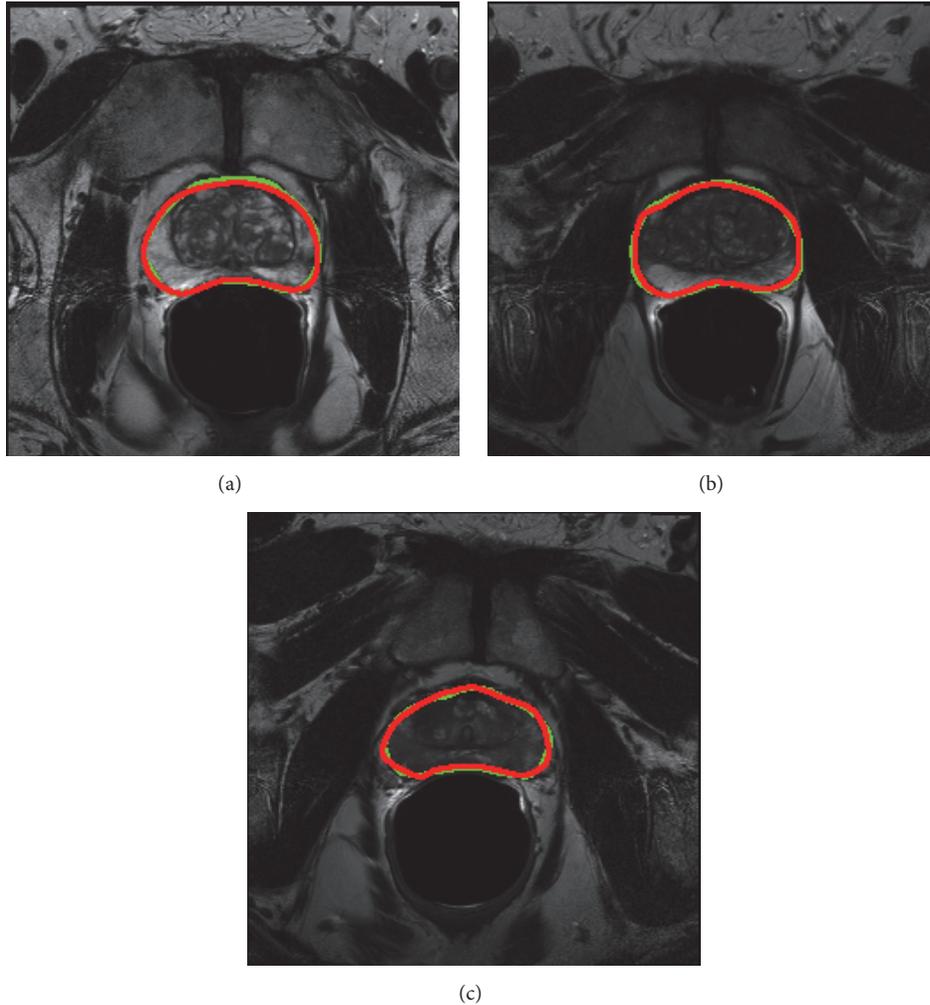


FIGURE 13: Parts of MRI prostate images segmentation results. The red and green contour denote ground truth and the result of UR-Net.

distributions are inhomogeneous both inside and outside. In addition, both prostate and nonprostate regions in those images have similar intensity distributions as shown in Figure 14.

As presented in the third column, FCNs only can detect and segment a part of prostate. And the segmentation results are not accurate, due to the fact that the FCNs model has assigned the labels to a small patch rather than each pixel. Besides, the FCNs ignore the boundaries information. So the FCNs model cannot be directly used in prostate segmentation problem.

As shown in the fourth column, U-Net model has got more accurate segmentation results than FCNs. Because U-Net assigns each label to every pixel and the architecture of U-Net can enhance information propagation through the whole network and improves the network performance, for the slices at the apex and base which lack clear boundary and complete texture, the model cannot segment the prostate accurately.

The results of modified FCNs are shown in the fifth column. Compared with original FCNs, the segmentation

results of modified FCNs are more accurate. From the results, we can see that the modified FCNs can detect more prostate information under the guidance of previous slice. The improvement of modified FCNs can be attributed to the superiority of the architecture of BDC-LSTM. Compared with the traditional convolutional layer, the BDC-LSTM layer can obtain the losing information from adjacent slices and enhance the performance of network.

The sixth column shows the results of V-Net. Compared with FCNs and U-Net, V-Net can take fully use of the 3D spatial information of the volumetric data. However, due to the limited data and memory, each time, V-Net only can receive local volume; this results in V-Net unable to obtain global information. From Figure 14, we can see that the prostate boundaries lose continuity and curvature.

The results of UR-Net are shown in the seventh column. We can observe that the model achieved the best results on prostate segmentation. It can be attributed to the fact that prostate sequence scans can provide more information than a single slice. And the model utilizes interslice information to aid the segmentation process.

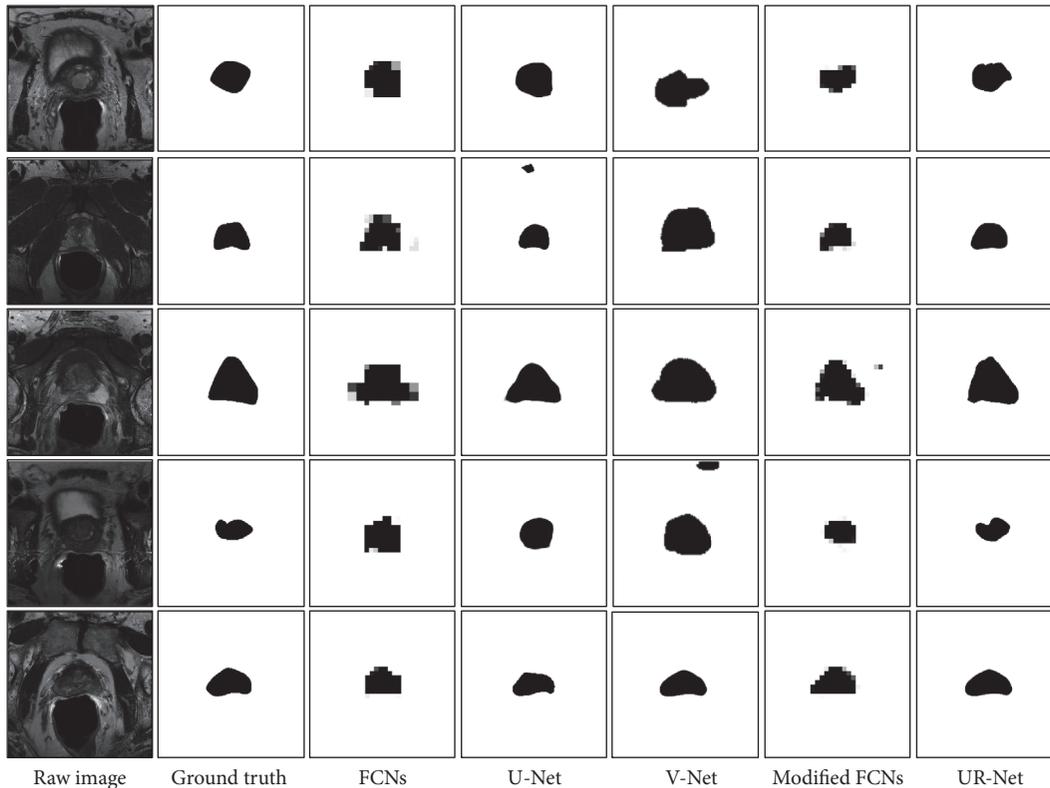


FIGURE 14: Segmentation results. From left to right are raw image, ground truth, the segmentation results of FCNs, U-Net, V-Net, modified FCNs, and UR-Net.

3.3.2. Quantitative Comparison. To quantitatively evaluate the segmentation results, we have computed segmentation results from three aspects as shown in Table 1 including the mean, maximum, and median DSC values. From Table 1, it can be seen that our proposed model obtained the highest scores among all the methods. It shows that the deep neural network with BDC-LSTM layers can obtain promising improvements on prostate MRI images segmentation. Besides, the modified FCNs obtain more accurate segmentation results compared with original FCNs. This improvement should be attributed to the superiority of BDC-LSTM layers, which utilize interslice as shape prior to guide the process of feature extraction and explore necessary information from interslice to alleviate information loss and finally improve the segmentation results.

4. Conclusions

In this paper, we propose a deep neural network with RNNs layers for MRI prostate image segmentation. Different from traditional methods, we treat the prostate scans as sequence data. Except for the local features, we also utilize the interslice information to aid prostate segmentation. In the proposed network, we put three neighboring slices into the network once. And then the network extracts intraslice contexts under the guidance of previous segmentation results from different neighboring slices. Connecting the two different features maps coming from opposite sequential directions together

TABLE 1: Quantitative evaluation results of the segmentation methods.

Methods	DSC values		
	Mean	Maximum	Median
UR-Net	0.9361	0.9772	0.9413
Modified FCNs	0.9047	0.9428	0.9143
V-Net	0.9180	0.9642	0.9380
U-Net	0.9197	0.9684	0.9402
FCNs	0.8489	0.9128	0.8571

can alleviate features lost. Experimental results on extensive MRI prostate image datasets demonstrate that the proposed model achieves better performance than the state-of-the-art convolutional neural networks.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under Grants U1536204, 60473023, and 61471274. And the authors would like to acknowledge

NVIDIA Corporation for the donation of the Titan Xp GPU used for this research.

References

- [1] L. Gong, S. D. Pathak, D. R. Haynor, P. S. Cho, and Y. Kim, "Parametric Shape Modeling Using Deformable Superellipses for Prostate Segmentation," *IEEE Transactions on Medical Imaging*, vol. 23, no. 3, pp. 340–349, 2004.
- [2] J. Dowling, J. Fripp, P. Greer, S. Ourselin, and O. Salvado, "Automatic atlas-based segmentation of the prostate: a miccai 2009 prostate segmentation challenge entry," in *Medical Image Computing and Computer-Assisted Intervention*, 2009.
- [3] J. Wu, Z. Cai, and Z. Gao, "Dynamic K-nearest-neighbor with distance and attribute weighted for classification," in *Proceedings of the 2010 International Conference on Electronics and Information Engineering, ICEIE 2010*, pp. V1356–V1360, Kyoto, Japan, August 2010.
- [4] J. Wu, Z. Cai, and X. Zhu, "Self-adaptive probability estimation for Naive Bayes classification," in *Proceedings of the 2013 International Joint Conference on Neural Networks, IJCNN 2013*, pp. 1–8, Dallas, TX, USA, August 2013.
- [5] J. Wu, S. Pan, X. Zhu, C. Zhang, and P. S. Yu, "Multiple Structure-View Learning for Graph Classification," *IEEE Transactions on Neural Networks and Learning Systems*, vol. PP, no. 99, pp. 1–16, 2017.
- [6] R. Toth, P. Tiwari, M. Rosen et al., "A magnetic resonance spectroscopy driven initialization scheme for active shape model based prostate segmentation," *Medical Image Analysis*, vol. 15, no. 2, pp. 214–225, 2011.
- [7] M. Samiee, G. Thomas, and R. Fazel-Rezai, "Semi-automatic prostate segmentation of mr images based on flow orientation," *IEEE International Symposium on Signal Processing and Information Technology*, pp. 203–207, 2006.
- [8] S. Klein, "Automatic segmentation of the prostate in 3D MR images by atlas matching using localized mutual information," *Medical Physics*, vol. 35, no. 4, pp. 1407–1417, 2008.
- [9] Q. Dou, H. Chen, Y. Jin, L. Yu, J. Qin, and P.-A. Heng, "3D deeply supervised network for automatic liver segmentation from CT volumes," in *Medical Image Computing and Computer-Assisted Intervention*, 2016.
- [10] H. Fu, Y. Xu, S. Lin, D. W. K. Wong, and J. Liu, "DeepVessel: Retinal Vessel Segmentation via Deep Learning and Conditional Random Field," in *Medical Image Computing and Computer-Assisted Intervention*, 2016.
- [11] H. Yang, J. Sun, H. Li, L. Wang, and Z. Xu, "Deep fusion net for multi-atlas segmentation: Application to cardiac MR images," in *Medical Image Computing and Computer-Assisted Intervention*, 2016.
- [12] Q. Zhu, B. Du, B. Turkbey, P. L. Choyke, and P. Yan, "Deeply-supervised CNN for prostate segmentation," in *Proceedings of the 2017 International Joint Conference on Neural Networks, IJCNN 2017*, pp. 178–184, Anchorage, AK, USA, May 2017.
- [13] S. Bao and A. C. S. Chung, "Multi-scale structured CNN with label consistency for brain MR image segmentation," *Computer Methods in Biomechanics and Biomedical Engineering: Imaging and Visualization*, pp. 1–5, 2016.
- [14] K. Fritscher, P. Raudaschl, P. Zaffino, M. F. Spadea, G. C. Sharp, and R. Schubert, "Deep neural networks for fast segmentation of 3D medical images," in *Medical Image Computing and Computer-Assisted Intervention*, 2016.
- [15] H. Huang, X. Hu, Y. Zhao et al., "Modeling Task fMRI Data via Deep Convolutional Autoencoder," *IEEE Transactions on Medical Imaging*, 2017.
- [16] J. Lv, B. Lin, W. Zhang et al., "Task fmri data analysis based on supervised stochastic coordinate coding," *Medical Image Analysis*, vol. 38, pp. 1–16, 2017.
- [17] Y. Xie, Z. Zhang, M. Sapkota, and L. Yang, "Spatial clockwork recurrent neural network for muscle perimysium segmentation," in *Medical Image Computing and Computer-Assisted Intervention*, 2016.
- [18] J. Chen, L. Yang, Y. Zhang, M. S. Alber, and D. Z. Chen, "Combining fully convolutional and recurrent neural networks for 3D biomedical image segmentation," in *In NIPS Proceedings*, MIT Press, Cambridge, MA, USA, 2016.
- [19] X. Qin, X. Li, Y. Liu, H. Lu, and P. Yan, "Adaptive shape prior constrained level sets for bladder MR image segmentation," *IEEE Journal of Biomedical and Health Informatics*, vol. 18, no. 5, pp. 1707–1716, 2014.
- [20] R. Dipietro, C. Lea, A. Malpani et al., "Recognizing surgical activities with recurrent neural networks," in *Medical Image Computing and Computer-Assisted Intervention*, 2016.
- [21] R. P. K. Poudel, P. Lamata, and G. Montana, "Recurrent fully convolutional neural networks for multi-slice MRI cardiac segmentation," in *Medical Image Computing and Computer-Assisted Intervention*, 2016.
- [22] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.
- [23] A. Graves, A.-R. Mohamed, and G. Hinton, "Speech recognition with deep recurrent neural networks," in *Proceedings of the 38th IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '13)*, pp. 6645–6649, May 2013.
- [24] J. T. Connor, R. D. Martin, and L. E. Atlas, "Recurrent neural networks and robust time series prediction," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 5, no. 2, pp. 240–254, 1994.
- [25] A. Graves, "Generating sequences with recurrent neural networks," 2013, arXiv preprint arXiv:1308.0850.
- [26] P. J. Werbos, "Backpropagation through time: what it does and how to do it," *Proceedings of the IEEE*, vol. 78, no. 10, pp. 1550–1560, 1990.
- [27] R. J. Williams and D. Zipser, "A learning algorithm for continually running fully recurrent neural networks," *Neural Computation*, vol. 1, no. 2, pp. 270–280, 1989.
- [28] J. H. Lee and N. L. Ricker, "Extended Kalman filter based nonlinear model predictive control," *Industrial & Engineering Chemistry Research*, vol. 33, no. 6, pp. 1530–1541, 1994.
- [29] A. J. Robinson and F. Fallside, "The utility driven dynamic error propagation network," Technical Report CUED/F-INFENG/TR.1, Cambridge University Engineering Department, 1987.
- [30] O. Ronneberger, P. Fischer, and T. Brox, "U-net: convolutional networks for biomedical image segmentation," in *Medical Image Computing and Computer-Assisted Intervention*, pp. 234–241, 2015.
- [31] F. Milletari, N. Navab, and S.-A. Ahmadi, "V-Net: fully convolutional neural networks for volumetric medical image segmentation," in *Proceedings of the 4th International Conference on 3D Vision (3DV '16)*, pp. 565–571, IEEE, October 2016.
- [32] P. Charles, Project title. GitHub repository (2013).

Research Article

Transferable Feature Representation for Visible-to-Infrared Cross-Dataset Human Action Recognition

Yang Liu ^{1,2}, Zhaoyang Lu,^{1,2} Jing Li,^{1,2} Chao Yao ³, and Yanzi Deng^{1,2}

¹School of Telecommunications Engineering, Xidian University, No. 2 South Taibai Road, Xi'an, Shaanxi 710071, China

²State Key Laboratory of Integrated Services Networks, Xidian University, No. 2 South Taibai Road, Xi'an, Shaanxi 710071, China

³School of Automation, Northwestern Polytechnical University, No. 127, West Youyi Road, Xi'an, Shaanxi 710072, China

Correspondence should be addressed to Chao Yao; yaochao@nwpu.edu.cn

Received 29 June 2017; Revised 5 January 2018; Accepted 9 January 2018; Published 15 February 2018

Academic Editor: Shirui Pan

Copyright © 2018 Yang Liu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Recently, infrared human action recognition has attracted increasing attention for it has many advantages over visible light, that is, being robust to illumination change and shadows. However, the infrared action data is limited until now, which degrades the performance of infrared action recognition. Motivated by the idea of transfer learning, an infrared human action recognition framework using auxiliary data from visible light is proposed to solve the problem of limited infrared action data. In the proposed framework, we first construct a novel Cross-Dataset Feature Alignment and Generalization (CDFAG) framework to map the infrared data and visible light data into a common feature space, where Kernel Manifold Alignment (KEMA) and a dual aligned-to-generalized encoders (AGE) model are employed to represent the feature. Then, a support vector machine (SVM) is trained, using both the infrared data and visible light data, and can classify the features derived from infrared data. The proposed method is evaluated on InfAR, which is a publicly available infrared human action dataset. To build up auxiliary data, we set up a novel visible light action dataset XD145. Experimental results show that the proposed method can achieve state-of-the-art performance compared with several transfer learning and domain adaptation methods.

1. Introduction

Human action recognition aims to recognize an ongoing action from a video clip, which has received great attention in recent years due to its wide applications, including video surveillance [1], video labeling [2], video content retrieval [3], human-computer interaction [4], and sports video analysis [5]. Over the past decades, significant progress has been made in action recognition [6] and most of the state-of-the-art approaches for action recognition have been put into visible imaging videos [7–9]. In addition, many visible light action datasets have been constructed for action recognition, such as KTH [10], HMDB51 [11], and UCF101 [12]. Generally speaking, human action recognition in visible light has been well addressed and successfully applied to some applications. However, illumination change, shadow, background clutter, and occlusion remain to be great challenges for visible light action recognition [13].

With the development of sensor technology, human actions can be captured by the thermal infrared cameras instead of the visible light ones. Compared with visible light action recognition, infrared action recognition can solve the aforementioned challenges [14]. For example, the infrared thermal imaging is robust to illumination change because it can capture humans well under poor light condition when the person can almost not be seen in the visible light videos, which is very useful for night surveillance or human-computer interaction (HCI) under dim night. In addition, as the temperatures of the shadow, background clutter, and occlusion are relatively low compared with that of the humans or moving objects in infrared videos, these challenges can be well suppressed in infrared videos. With these properties, infrared action recognition can be adopted in more applications and outperform that in visible light. Therefore, infrared action recognition may become a next hot topic in computer vision in the future.

Actually, infrared and visible action data lie in different feature space and the traditional approaches for visible light action recognition cannot be directly applied to infrared action recognition due to the modality gap between them. However, the methods for infrared action recognition are limited. Furthermore, there is only one publicly available infrared dataset InfAR [20] for action recognition until now. As a result, the performance of infrared action recognition in previous works is preliminary and leaves a reasonable space to further promote its performance. To these issues, if a large amount of previously annotated videos from various visible light videos datasets can be transferred to infrared domains for recognition, considerable amount of time-consuming human efforts, such as collecting and hand labeling a large amount of infrared action videos, can be saved. In addition, as infrared and visible light videos may contain complementary information, infrared action recognition performance can be improved if the knowledge from visible light and infrared data can be properly integrated. Nevertheless, it would have at least two obstacles to integrate these two datasets. (1) Infrared and visible light videos are captured by different sensors; the strong modality gap between them will degrade the recognition performance without effective transferable feature representation. (2) In real-world scenario, infrared videos are usually limited while the visible ones are abundant. The imbalanced data distribution will also degrade the classification performance.

Tackling these problems, a novel Cross-Dataset Feature Alignment and Generalization (CDFAG) framework is proposed for infrared action recognition task in this paper. To be more specific, we focus on adapting, aligning, and generalizing representations from different domains to a single common feature space in order to bring the original target domain (infrared action data) and the auxiliary source domain (visible light action data) into the same feature space. And then we learn a unique classifier in that semantically meaningful aligned and generalized feature space across datasets. In this way, the modality gap between these two datasets is reduced. To better use the data in the generalized feature space, we adopt semisupervised technique so that both the labeled and unlabeled data are considered in our method. In more detail, Kernel Manifold Alignment (KEMA) [15] is adapted to cross-dataset action recognition to generate aligned representations and then cross-domain generalized features are learnt by training two novel aligned-to-generalized encoders (AGE) on the source and target datasets in parallel. To build up source domain data, we set up a new visible light action dataset called XD145. Putting all the things together, we can summarize the main contributions of this paper as follows:

- (i) We have proposed a novel Cross-Dataset Feature Alignment and Generalization (CDFAG) framework to address the visible-to-infrared action recognition problem. It can efficiently reduce the modality gap across datasets and generate aligned and generalized feature representations in a common space with low intra-class diversity and high inter-class variance.
- (ii) It is the first time Kernel Manifold Alignment (KEMA) is applied in infrared action recognition

field to generate aligned representations in a common latent space.

- (iii) We have designed a novel aligned-to-generalized encoder (AGE) model to learn generalized feature representations after feature alignment by KEMA.
- (iv) We achieved state-of-the-art results in visible-to-infrared action recognition compared with several transfer learning, domain adaptation, and deep learning based methods.
- (v) Since there are a limited number of action videos from existing benchmark visible light datasets which share the same class label with that of the InfAR dataset, we have constructed a new visible light action dataset called XD145 to build up auxiliary source domain data. And this dataset could be further utilized as the benchmark visible light action dataset.

The rest of the paper is organized as follows. In Section 2, we review some background and related works. In Section 3, we explain details of our proposed method. Section 4 presents the experimental results of our proposed method on visible-to-infrared action recognition and cross-dataset action recognition, and finally Section 5 draws the conclusion and future research lines.

2. Related Work

In this section, we present the background and related works. We briefly review the concepts, methods in transfer learning and domain adaptation, their benefits in cross-domain action recognition, and the development status of infrared action recognition.

2.1. Transfer Learning and Domain Adaptation. The classical pattern recognition and machine learning tasks [21–23] mainly adopt a robust classifier learnt by annotated training data and assume the testing data and the training data belong to the same feature space or distribution. However, it is unrealistic in real-world applications because of the high price of manual labeling training samples and environmental restrictions. Therefore, sufficient training data that share the same feature space or distribution with the testing data cannot always be guaranteed even using some feature selection methods [24, 25] without considering the distribution gap. In this case, the potential discriminability of the trained model can be limited by the insufficient training data. Of the several schools of thought addressing this problem, two prominent ones are transfer learning [26] and domain adaptation [27]. In fact, transfer learning methods are closely related but not equivalent to domain adaptation. Transfer learning aims to transfer the knowledge from a source domain to the target domain while domain adaptation methods are essentially solving transfer learning problems. Surveys like [26] show that the type of knowledge being transferred can be roughly classified into four categories: (1) instance transfer, (2) feature-representation transfer, (3) parameter transfer, and (4) relational-knowledge transfer. Our proposed method falls

into feature-representation transfer by adapting the representations from different domains to a single common latent space. The literature of feature-representation transfer can be roughly divided into three kinds of adaptation problems: supervised, unsupervised, and semisupervised adaptation problems, depending on the availability of labels in different domains.

Semisupervised domain adaptation has attracted much attention in recent years. For example, a reconstruction-based domain adaptation method called latent sparse domain transfer (LSDT) was proposed in [28] for visual categorization of heterogeneous data via subspace learning and sparse representation. A $l_{2,1}$ -norm based discriminative robust kernel transfer learning (DKTL) method was proposed in [29] to address distribution mismatch problem of image classification across domains. Although the methods in [28, 29] achieve good performance in cross-domain image classification, to the best of our knowledge, a more challenging problem of visible-to-infrared cross-dataset action recognition was not studied. Extreme learning machine was used in [30] to address the visual knowledge adaptation problem for video event recognition and object recognition. It should be noted that the proposed method is based on feature-representation domain adaption, which is essentially different from the extreme learning machine methods, which are classifier-based domain adaptation approaches. Apart from the methods mentioned above, manifold alignment is an important kind of semisupervised domain adaptation methods, which concurrently matches the corresponding samples and preserves the geometry of each domain by graph Laplacian [31]. Actually, data manifolds alignment boils down to finding projections to a common latent space. Semisupervised method Kernel Manifold Alignment (KEMA) was proposed in [15] and has been successfully applied to multimodal visual object recognition [15], multisubject facial expression recognition [15], and multitemporal remote sensing image classification [32]. Nevertheless, Kernel Manifold Alignment (KEMA) method has not been applied in cross-domain human action recognition. Therefore, we have studied the effectiveness of the Kernel Manifold Alignment (KEMA) and adapted it to visible-to-infrared action recognition to obtain aligned feature representations across datasets. The main focus of our work is on the use of KEMA for feature alignment in visible-to-infrared action recognition which was not addressed in [15].

2.2. Cross-Domain Action Recognition via Transfer Learning.

With the development of action recognition, applying transfer learning to action recognition datasets generated by different sensors such as visible light cameras, infrared cameras, RGB-D cameras, wearable sensors, or other sensor modalities has received great interests in recent years [33]. As video sequences are the most common type of action datasets, the difference in manipulating and deploying the camera to capture actions leads to different issues, for example, various camera viewing angles, cluttered background, illumination changes, and different light spectrums such as visible light spectrum and infrared spectrum, and all contribute to

significant variance in the captured videos. Therefore, action recognition especially cross-domain action recognition is a challenging problem.

Transferring knowledge for cross-view action recognition is prevailing [34]. For example, Zheng et al. [35] proposed learning a pair of dictionaries simultaneously from videos pairs taken at different views to encourage each video pair to have the same sparse representation. Zhang et al. [36] proposed a linear transformation to transform source view to target view via virtual path. Wu et al. [37] proposed a method to discover a discriminative common representation space where source and target views are linked and knowledge is transferred between them. Sui et al. [38] introduced two different projection matrices to map the action data from two different views into the common space with low intraclass diversity and high interclass variance and reducing the mismatch between them. Zu and Zhang [39] introduced a method called Canonical Sparse Cross-view Correlation Analysis to address multiview feature extraction problem. Different from the above-mentioned cross-view action recognition problems, we tackle the visible-to-infrared action recognition problem, which is essentially a cross-dataset action recognition problem.

In recent works about cross-dataset action recognition, Bian et al. [40] proposed a transfer topic model (TTM) which utilized information from the auxiliary domain to assist recognition tasks in the target domain. Zhu and Shao [18] introduced a weakly supervised cross-domain dictionary learning (WSCDDL) approach which learns a reconstructive, discriminative, and domain-adaptive dictionary pair and the corresponding classifier parameters to address cross-domain image classification, action recognition, and event recognition problems. Tang et al. [41] improved the accuracy of action recognition in RGB videos by activating the borrowing of visual knowledge across different video modalities such as RGB videos, the depth maps, and the skeleton data of actions. Liu et al. [42] proposed a simple to complex action transfer learning model (SCA-TLM) to leverage the abundant labeled simple actions to improve the performance of complex action recognition. Although these works can achieve promising results in their related fields, visible-to-infrared action recognition has not attracted much attention until now.

With the revival of neural networks in recent years, many neural networks based transfer learning methods have been proposed as well. Kan et al. [43] proposed bishifting autoencoder network (BAE) to alleviate the discrepancy between source and target domains and evaluate its effectiveness in face recognition. Xu et al. [19] tackled the cross-dataset action recognition problem by training a pair of many-to-one encoders in parallel to map raw features from the source and target datasets to the same space. Although dual many-to-one encoder in [19] can generalize features well across datasets, it requires a large number of labeled training samples from both source and target datasets to learn domain-invariant features without utilizing auxiliary domain data as an aide. In addition, the inputs of the encoders are raw action features named action bank features without considering feature alignment in advance. Different from the

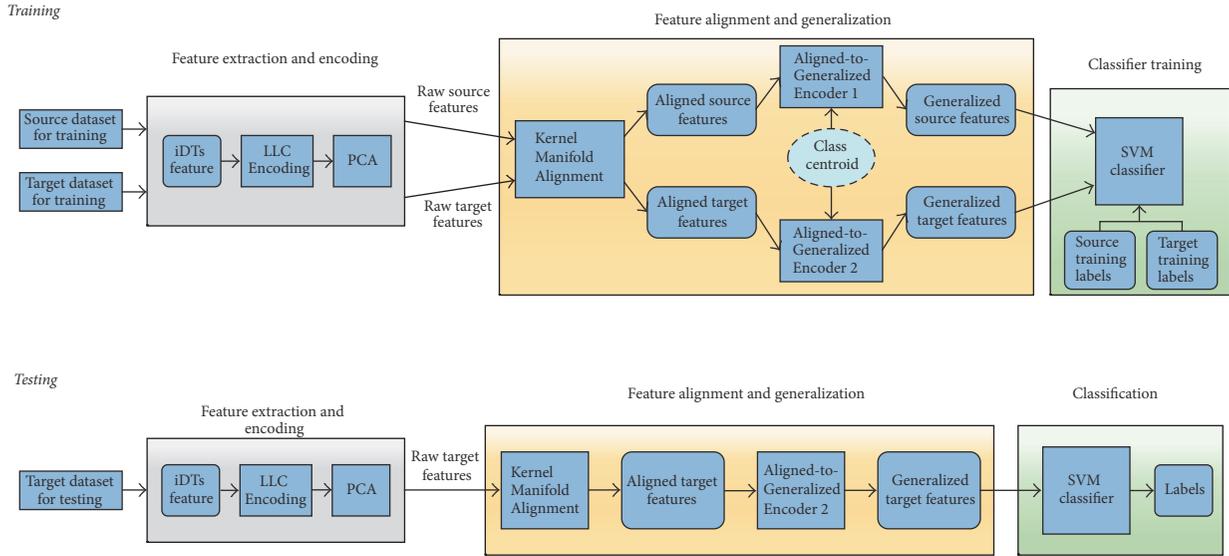


FIGURE 1: The framework of cross-dataset feature alignment and generalization (CDFAG). Top: training. Bottom: testing.

above-mentioned neural networks based transfer learning methods, we take both feature alignment and auxiliary domain data into consideration and propose a novel aligned-to-generalized encoder (AGE) model to map the aligned feature representations to the same generalized feature space with low intra-class diversity and high inter-class variance.

2.3. Infrared Action Recognition. Basically, most of the current research efforts for action recognition have been put in visible light videos while infrared action recognition has not attracted much attention. Recently, increasing efforts have been devoted to infrared action recognition. For example, Han and Bhanu [44] proposed an efficient spatiotemporal representation for human repetitive action recognition under thermal infrared scenarios. Han and Bhanu [14] introduced a hierarchical scheme to combine color and thermal images to improve human silhouette detection. Eum et al. [45] used hog and support vector machine to realize infrared action recognition at night. However, these works focus on simple actions under relatively simple environment with limited infrared data. In addition, there is no publicly available infrared action recognition dataset until Gao et al. [13] built the first public infrared action recognition dataset called InfAR. In [13], state-of-the-art action recognition pipelines including widely used low-level local descriptors were evaluated in InfAR dataset. Then Gao et al. [20] extended their previous work [13] and utilized several state-of-the-art pipelines based on low-level features and deep convolutional neural network to evaluate their new infrared action recognition dataset (InfAR). However, the best recognition accuracy in [20] is relatively low and leaves a reasonable space to further promote the performance on InfAR dataset.

Actually, transfer learning has seldom been applied to infrared action recognition. For example, Zhu and Guo [46] proposed applying adaptive support vector machine (A-SVM) [47] to adapt the existing visible light action classifier

to classify infrared actions and achieved preliminary results in their own dataset. Although the adaptive support vector machine (A-SVM) based method in [46] can perform better than direct matching, A-SVM is essentially a classifier-transfer based method without considering the max-margin property for the adapted classifier on target instances; therefore it faces accuracy degradation as a result of overfitting.

Our proposed method differs from the above-mentioned approaches in such aspects that it more comprehensively projects and aligns data from different domains in a nonlinear way through kernelization to generate aligned representations and learns a dual aligned-to-generalized encoders to obtain cross-domain generalized features while considering both the discriminability and domain adaptability at the same time. In our proposed CDFAG, the learned classifier across source domain and target domain becomes more discriminative against modality gap because of the integration of both source and target domain knowledge, while a majority of previous transfer learning methods focus on incomplete target domain without utilizing other domain data as an aide to improve the performance of original recognition systems. We will detail our proposed method in Section 3.

3. The Proposed Method

In this section, we detail our proposed CDFAG. An overview of the CDFAG is presented in Figure 1. Actually, the proposed CDFAG is semisupervised as both the labeled and the unlabeled data are used in source and target training sets. Our proposed CDFAG consists of three stages. In the first stage, feature extraction and encoding are accomplished on both the source and target datasets, where improved dense trajectories (iDTs) features are extracted, encoded, and reduced to a low-dimensional subspace. In the second stage, aligned features of source and target domains are generated by Kernel Manifold Alignment, then a pair of aligned-to-generalized encoders are

trained on the source and target datasets in parallel guided by the centroids of training aligned instances from each class, and after that the output values of the encoders are extracted as the ultimate generalized representations. Finally, a support vector machine is built on the generalized features extracted from both the source and target datasets and then used to classify the new features extracted from unseen samples of the target dataset.

3.1. Preprocessing

3.1.1. Feature Extraction and Encoding. In this paper, we choose improved dense trajectories (iDTs) [48] features with trajectory shape, HOF, MBHx, and MBHy as the low-level action video representation. The total length of the feature vector is 330. Specifically, we use the implementation released on the website of Wang (https://lear.inrialpes.fr/people/wang/improved_trajectories/) for iDTs and choose the default parameter setting. For iDTs, a large number of local trajectory descriptors may lead to high computational complexity and memory consumption. To cope with this issue, we adopt Locality-constrained Linear Coding (LLC) [49] scheme to represent the iDTs by multiple bases, which can bring less quantization error while preserving the local smooth sparsity. Taking both efficiency and the construction error into consideration, LLC encoding scheme is applied to the iDTs with 5 local bases, and the codebook size is set to be 4000 for all training-testing partitions. Thus, the dimension of the encoded iDTs features is 4000. To reduce the complexity when constructing the codebook, only 200 local iDTs are randomly selected from each video sequence.

3.1.2. Principal Component Analysis. After LLC encoding, the feature representations are still high dimensional and strongly correlated. To obtain more compacted feature representation, we utilize principal component analysis (PCA) [50] to preprocess these features. In our method, we retain top p principal components such that the cumulative corresponding eigenvalues cover over 99% of the total eigenvalues. In our experiments, this reduces feature dimension down to the range of 500 to 600, varying between datasets.

3.2. Feature Alignment by Kernel Manifold Alignment. In this section, we detail the feature alignment method based on Kernel Manifold Alignment (KEMA). An illustration of how feature alignment functions is shown in Figure 2.

3.2.1. Notation. To fix notation, we consider K input domains. The data instances of each domain belong to c different classes. Let $X_k = (x_k^1, \dots, x_k^{m_k})$ represent the k th input domain, where m_k is the number of samples in the k th domain. The idea of kernelization is to map the input data instance into a high dimensional Hilbert space \mathcal{H} with the mapping function $\phi : \mathbf{x} \mapsto \phi(\mathbf{x}) \in \mathcal{H}$ such that the mapped data is better suited for solving our problem linearly. Kernel trick is adopted in our method to avoid high computational load. Therefore, we define a kernel function $\mathbf{K}_{ij} = K(\mathbf{x}_i, \mathbf{x}_j) = \langle \phi(\mathbf{x}_i), \phi(\mathbf{x}_j) \rangle_{\mathcal{H}}$ computing the similarity between mapped instances without having to compute $\phi(\cdot)$ explicitly. Many

common types of kernel functions can be adopted in KEMA, such as the RBF kernel, the linear kernel, and the polynomial kernel. In this paper, we use the RBF kernel. Considering multiple data modalities here, we would have to map the K datasets to K Hilbert spaces \mathcal{H}_k of dimension H_k , $\phi_k(\cdot) : \mathbf{x} \mapsto \phi_k(\mathbf{x}) \in \mathcal{H}_k$, $k = 1, \dots, K$.

3.2.2. Kernel Manifold Alignment (KEMA). The KEMA method aims to construct K domain-specific projection functions, f_1, f_2, \dots, f_K , to project the data in Hilbert space from all K domains to a new common latent space, on which the instances' topology of each domain is preserved, the instances from the same classes will locate nearby, and the ones from different classes will be far from each other. To do so, KEMA aims to find a data projection matrix $F = [f_1, f_2, \dots, f_K]^T$ that minimizes the following cost function:

$$\begin{aligned} \{f_1, f_2, \dots, f_K\} &= \arg \min_{f_1, f_2, \dots, f_K} (C(f_1, f_2, \dots, f_K)) \\ &= \arg \min_{f_1, f_2, \dots, f_K} \left(\frac{\mu \text{TOP} + (1 - \mu) \text{SIM}}{\text{DIS}} \right), \end{aligned} \quad (1)$$

where TOP, SIM, and DIS denote the topology, class similarity, and class dissimilarity, respectively. μ is a parameter balancing the contribution of the similarity and the topology terms. As $\mu \in [0, 1]$, we can see that when $\mu > 0.5$, more importance is given to topology and vice versa. The three terms are defined as follows:

- (1) Minimizing a topology-preservation term, TOP, which aims to preserve the local topology of each data domain:

$$\begin{aligned} \text{TOP} &= \sum_{k=1}^K \sum_{i,j=1}^{m_k} \left\| \mathbf{f}_k^T \phi_k(\mathbf{x}_k^i) - \mathbf{f}_k^T \phi_k(\mathbf{x}_k^j) \right\|^2 W_t^k(i, j) \\ &= \text{tr}(\mathbf{F}^T \Phi \mathbf{L}_t \Phi^T \mathbf{F}), \end{aligned} \quad (2)$$

where W_t^k in the similarity matrix representing the similarity of x_k^i and x_k^j , which can be computed as $e^{-\|x_k^i - x_k^j\|^2}$. $L_t \in \mathcal{R}^{(\sum_k m_k) \times (\sum_k m_k)}$ is the graph Laplacian matrix issued from $L_t = D_t - W_t$ while D_t is the diagonal row sum matrix defined as $D_t(i, i) = \sum_j W_t(i, j)$.

- (2) Minimizing a class similarity term, SIM, which encourages the locations of instances with the same class label to be close with each other in the new latent space:

$$\begin{aligned} \text{SIM} &= \sum_{k,k'=1}^K \sum_{i,j=1}^{m_k, m_{k'}} \left\| \mathbf{f}_k^T \phi_k(\mathbf{x}_k^i) - \mathbf{f}_{k'}^T \phi_{k'}(\mathbf{x}_{k'}^j) \right\|^2 W_s^{k,k'}(i, j) \\ &= \text{tr}(\mathbf{F}^T \Phi \mathbf{L}_s \Phi^T \mathbf{F}), \end{aligned} \quad (3)$$

where $W_s^{k,k'}$ in the similarity matrix is set to be 1 if two instances from domains k and k' share the

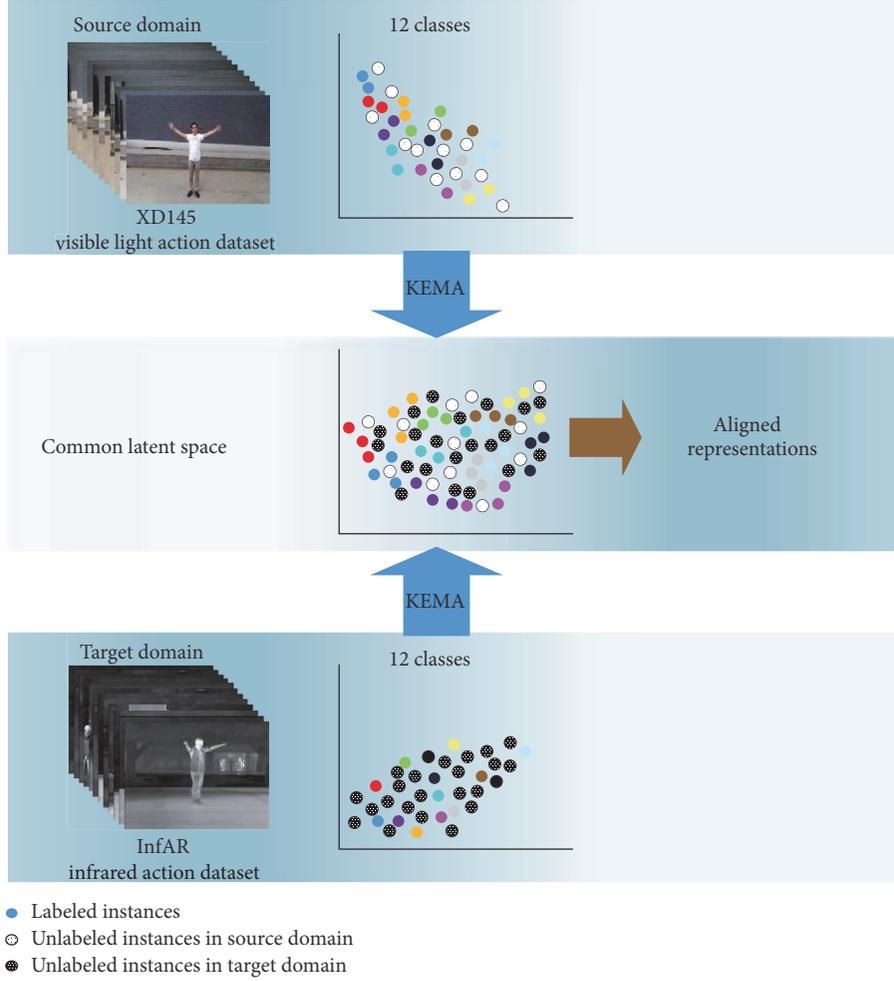


FIGURE 2: Illustration of feature alignment in a visible-to-infrared action recognition setting. Circles in different colors represent different action classes.

same class label and 0 otherwise (including the case when the label information is not available). The corresponding diagonal row sum matrix is defined as $D_s(i, i) = \sum_j W_s(i, j)$ and the graph Laplacian matrix $L_s = D_s - W_s$.

- (3) Maximizing a class dissimilarity term, DIS, which encourages instances with different class labels to be separated in the new latent space,

DIS

$$= \sum_{k, k'=1}^K \sum_{i, j=1}^{m_k, m_{k'}} \left\| \mathbf{f}_k^\top \phi_k(\mathbf{x}_k^i) - \mathbf{f}_{k'}^\top \phi_{k'}(\mathbf{x}_{k'}^j) \right\|^2 W_d^{k, k'}(i, j) \quad (4)$$

$$= \text{tr}(\mathbf{F}^\top \Phi \mathbf{L}_d \Phi^\top \mathbf{F}),$$

where $W_d^{k, k'}$ in the dissimilarity matrix is set to be 1 if two instances from domains k and k' are from different classes and 0 otherwise (including the case when the label information is not available). The corresponding diagonal row sum matrix is defined as

$$D_d(i, i) = \sum_j W_d(i, j) \text{ and the graph Laplacian matrix } L_d = D_d - W_d.$$

Given (2)–(4), the optimization problem is formalized as follows:

$$\arg \min_{f_1, f_2, \dots, f_K} \text{tr} \left(\frac{\mathbf{F}^\top \Phi (\mu \mathbf{L}_t + (1 - \mu) \mathbf{L}_s) \Phi^\top \mathbf{F}}{\mathbf{F}^\top \Phi \mathbf{L}_d \Phi^\top \mathbf{F}} \right). \quad (5)$$

It is straightforward that the solution of (5) boils down to finding the n lowest eigenvalues of the following generalized eigenvalue decomposition [51]:

$$\Phi (\mu \mathbf{L}_t + (1 - \mu) \mathbf{L}_s) \Phi^\top \mathbf{F} = \lambda \Phi \mathbf{L}_d \Phi^\top \mathbf{F}, \quad (6)$$

where Φ is a matrix containing the matrices $\Phi_k = [\phi_k(\mathbf{x}_k^1), \dots, \phi_k(\mathbf{x}_k^{m_k})]^\top$ in a block diagonal form and \mathbf{F} contains the row eigenvectors for the particular domain defined in Hilbert space \mathcal{H}_k , where $\mathbf{F} = [\mathbf{f}_1, \mathbf{f}_2, \dots, \mathbf{f}_H]^\top$, $H = \sum_{k=1}^K H_k$, and λ is the eigenvalues of the generalized eigenvalue decomposition problem. Note that Φ and \mathbf{F} are high dimensional and cannot be explicitly computed. Therefore,

the eigenvectors are expressed as a linear combination of mapped instances using the Riesz representation theorems [52] and $\mathbf{f}_k = \Phi_k \alpha_k$ and in matrix notation $\mathbf{F} = \Phi \Lambda$. In (6), by multiplying both sides by Φ^\top and replacing the dot products with the corresponding kernel matrices, $\mathbf{K}_k = \Phi_k^\top \Phi_k$, the final solution is obtained:

$$\mathbf{K}(\mu \mathbf{L}_t + (1 - \mu) \mathbf{L}_s) \mathbf{K} \Lambda = \lambda \mathbf{K} \mathbf{L}_d \mathbf{K} \Lambda, \quad (7)$$

where \mathbf{K} is a matrix containing the kernel matrices \mathbf{K}_k in a block diagonal form. The block structure of projection matrix Λ is as follows:

$$\Lambda = \begin{bmatrix} \alpha_1 \\ \vdots \\ \alpha_K \end{bmatrix} = \begin{bmatrix} \alpha_{1,1} & \cdots & \alpha_{1,n} \\ \vdots & \ddots & \vdots \\ \alpha_{n_1,1} & \cdots & \alpha_{n_1,n} \\ \alpha_{n_1+1,1} & \cdots & \alpha_{n_1+1,n} \\ \vdots & \ddots & \vdots \\ \alpha_{n,1} & \cdots & \alpha_{n,n} \end{bmatrix}, \quad (8)$$

where the eigenvectors for the first domain are highlighted in bold.

Once the projection matrix Λ is obtained, the instance \mathbf{x}_k^i from k th domain can be projected to the new latent space by first mapping \mathbf{x}_k^i to its corresponding kernel form \mathbf{K}_k^i and then applying the corresponding projection vector α_k defined therein:

$$P(\mathbf{x}_k^i) = \mathbf{f}_k^\top \Phi_k^i = \alpha_k^\top \Phi_k^\top \Phi_k^i = \alpha_k^\top \mathbf{K}_k^i, \quad (9)$$

where \mathbf{K}_k^i is a kernel evaluations vector between instance \mathbf{x}_i and all instances from k th domain used to define the projections α_k . Similar to eigenvalue decomposition based methods, the data can be projected onto a lower-dimensional subspace by simply preserving the first p columns of α_k , where $n = \sum_k n_k$ is the total number of samples involved in the kernel matrices and $p \ll n$. In this sense, KEMA leaves some control on the dimensionality of the latent space for feature alignment.

In this paper, the number of input domains K is set to 2 because there is only one target domain (infrared dataset) and one source domain (visible light dataset) in our experiments.

3.3. Feature Generalization by Aligned-to-Generalized Encoders. Due to the huge modality gap between infrared and visible light data, a unified subspace may not exist when only using KEMA to align features from both domains. To be more specific, KEMA holds the assumption that a unified aligned space for both the source and the target domain exists. This assumption is too strict and may be invalid for some cases. Therefore, we relax this strict assumption and learn transferable feature representations across infrared and visible domains in a hierarchical way. With the obtained aligned representation, aligned-to-generalized encoders (AGE) model is adopted to force the outputs to be identical to the input aligned instances from the same action

class. The AGE is trained by the guidance of the identical representation of the same action class, where intraclass diversities are minimized and generalized representations are generated across datasets. In this section, we present the architecture and details of the proposed AGE.

3.3.1. Target Output Generation. The centroid of each action class is used as the target output, which is computed by averaging over instances' aligned feature representations in each class. Let $X_{s,c}^i$ and $X_{t,c}^j$ denote the aligned representations of i th and j th training instances from class c in the source and target dataset, respectively, and the target output for instances from class c is defined as follows:

$$T_c = \frac{1}{N_{s,c} + N_{t,c}} \left(\sum_{i=1}^{N_{s,c}} X_{s,c}^i + \sum_{j=1}^{N_{t,c}} X_{t,c}^j \right), \quad (10)$$

where $N_{s,c}$ and $N_{t,c}$ denote the total number of instances of class c from the source and target datasets.

3.3.2. Encoders Training. At the feature generalization stage, a pair of aligned-to-generalized encoders are trained on the source and target datasets in parallel. For instances, with the same action class label, the target outputs of the two aligned-to-generalized encoders are identical, which forces the two aligned-to-generalized encoders to generalize to varying inputs and guide outputs of the same class instances to be similar. In this sense, aligned instances with the same class label from two datasets are mapped to the same feature space [36, 53] with low intraclass diversity [19] and then generalized and discriminative representations of the instances are generated across datasets.

In this section, we demonstrate the benefits of using a pair of aligned-to-generalized encoders for feature generalization in visible-to-infrared action recognition. The architecture of the aligned-to-generalized encoders (AGE) is illustrated in Figure 3. The AGE are essentially fully connected feedforward neural networks with an input layer, a hidden layer, and an output layer. Although the intuition that a deeper network architecture with more than one hidden layer can learn more robust and discriminative representations, it has been shown that carefully configured and trained single-hidden-layer neural networks can also achieve good performance in many tasks [19, 54], which is validated in our experiments as well. In addition, the architectures of the AGE trained on the source and target datasets are identical.

After feature extraction and encoding, the training videos from both datasets are represented by iDTs features encoded by LLC, which are then reduced to a low-dimensional subspace via PCA. Then, the KEMA method is applied to map the raw feature representations of instances from two datasets to the common latent space to obtain aligned features, shown in Figure 3. For simplicity, we denote one training aligned instance as x_i with dimension L . Therefore, both aligned-to-generalized encoders have an input layer of size L , a hidden layer of size H , and an output layer of size L , where L is the length of the ultimate aligned output feature vector and H and L are user defined parameters. In this work, we empirically

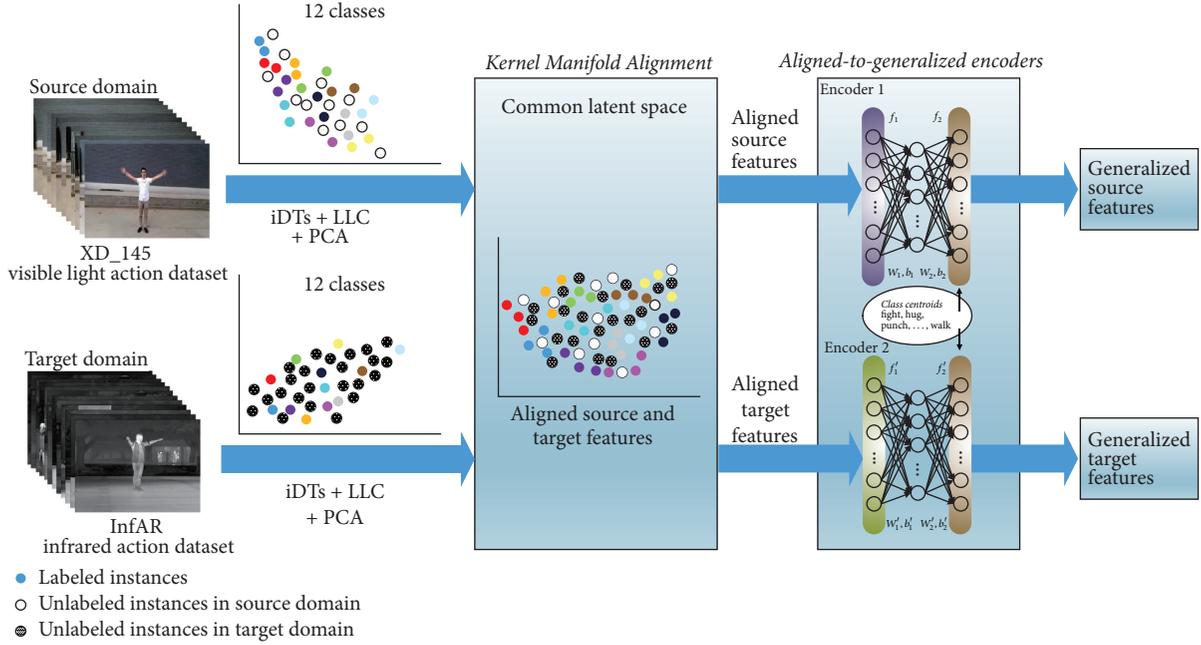


FIGURE 3: Illustration of aligned-to-generalized encoders generalizing aligned features across visible light and infrared action datasets.

restrict H to be equal to L and experimented with hidden layer sizes that range from 50 to 500 in Section 4.4.3.

Although both aligned-to-generalized encoders have the same architecture, they have different parameters. The goal of training the two encoders in parallel is to find a mapping between training aligned instances and target generalized outputs. Take the source encoder as an example; the mapping is accomplished via $f_1 : \mathbb{R}^L \rightarrow \mathbb{R}^H$ and $f_2 : \mathbb{R}^H \rightarrow \mathbb{R}^L$. f_1 and f_2 are defined as follows:

$$\begin{aligned} f_1(x_i) &= s(W_1 x_i + b_1), \\ f_2(f_1(x_i)) &= s(W_2 f_1(x_i) + b_2), \end{aligned} \quad (11)$$

where $s(\cdot)$ is the activation function and W_1, b_1, W_2 , and b_2 are the parameters for f_1 and f_2 , respectively. The encoder parameters are indicated in Figure 3. Given a hidden layer size H , the weights and biases in both encoders are initialized with random numbers drawn from a uniform distribution ranging between 0 and 1. The sigmoid function was used as the activation function:

$$s(x) = \frac{1}{1 + e^{-x}}. \quad (12)$$

The objective functions are defined as follows:

$$\begin{aligned} J &= \frac{1}{2N} \sum_{i=1}^N \|T_i - h_{W_1, b_1, W_2, b_2}(x_i)\|_2^2 \\ &= \frac{1}{2N} \sum_{i=1}^N \|T_i - f_2(f_1(x_i))\|_2^2 \\ &= \frac{1}{2N} \sum_{i=1}^N \|T_i - s(W_2 s(W_1 x_i + b_1) + b_2)\|_2^2, \end{aligned} \quad (13)$$

where $f_1(\cdot)$ and $f_2(\cdot)$ are defined in (11), W_1, W_2, b_1 , and b_2 are the weights and biases of the encoder, and N is the number of training instances.

Stochastic gradient descent was utilized to minimize the objective function J by iteratively updating the weights and biases. For example, W_1 is updated as follows:

$$\Delta W_1(t+1) = \eta \frac{\delta J_{t+1}}{\delta W_1} + \rho \Delta W_1(t), \quad (14)$$

$$W_1(t+1) = W_1(t) - \Delta W_1(t+1),$$

where $\Delta W_1(t+1)$ is the update to W_1 at the $(t+1)$ th iteration, J_{t+1} is the objective function value at the $(t+1)$ th iteration, η denotes the learning rate, and ρ denotes the momentum. W_2, b_1 , and b_2 are updated in a similar way.

In our method, the trained values of the output layer are extracted as the ultimate generalized features. When the objective function J is minimized, the output values of the encoder are approximate solutions staying close to the target outputs instead of being identical to the predefined target outputs. Therefore, the final features extracted from aligned instances of the same action class, from both the source and the target datasets, would lie in the same cluster with small intraclass diversity and high interclass variance. This phenomenon will be illustrated in Section 4.4.2.

To make it clear, the proposed CDFAG is summarized in Algorithm 1.

3.4. Classification. Due to the lack of infrared data, directly using neural network for classifying may lead to overfitting. Therefore, we just use the AGE as a feature extractor rather than a classifier to avoid overfitting. In our experiments, we use multiclass support vector machine (SVM) as a classifier

Input:

Raw features $X = \{X_k \mid X_k = (\mathbf{x}_k^1, \dots, \mathbf{x}_k^{m_k}), X_k \in \mathcal{R}^{d_k \times m_k}, k = 1, \dots, K\}$, the number of input data domains K , dimension of common latent subspace n , trade-off parameter $\mu = 0.1$, maximum iterations 1000, input layer size L , output layer size L , hidden layer size H , learning rate 0.1, momentum 0.9 and $W_1, W_1', b_1, b_1', W_2, W_2', b_2, b_2'$ are randomly initialized.

Feature alignment:

- (1) Map the raw features from K datasets to K Hilbert spaces:

$$\Phi = \{\Phi_k \mid \Phi_k = [\phi_k(\mathbf{x}_k^1), \dots, \phi_k(\mathbf{x}_k^{m_k})]^\top, k = 1, \dots, K\}.$$

- (2) Construct graph Laplacian matrices $\mathbf{L}_t, \mathbf{L}_s$ and \mathbf{L}_d defined in Section 3.2.2.

- (3) Compute the mapping functions (f_1, f_2, \dots, f_K) by finding the n smallest eigenvalues of the generalized eigenvalue problem:

$$\mathbf{K}(\mu \mathbf{L}_t + (1 - \mu) \mathbf{L}_s) \mathbf{K} \boldsymbol{\Lambda} = \lambda \mathbf{K} \mathbf{L}_d \mathbf{K} \boldsymbol{\Lambda}.$$

- (4) Apply (f_1, f_2, \dots, f_K) to map K input datasets to the new n dimensional common latent space to obtain aligned features:

$$P = \{P(\mathbf{x}_k^i) \mid P(\mathbf{x}_k^i) = \mathbf{f}_k^\top \Phi_k^i = \boldsymbol{\alpha}_k^\top \Phi_k^\top \Phi_k^i = \boldsymbol{\alpha}_k^\top \mathbf{K}_k^i, k = 1, \dots, K, i = 1, \dots, m_k\}.$$

Feature generalization:

- (5) Calculate the target outputs of aligned-to-generalized encoders from class c :

$$T_c = \frac{1}{N_{s,c} + N_{t,c}} \left(\sum_{i=1}^{N_{s,c}} X_{s,c}^i + \sum_{j=1}^{N_{t,c}} X_{t,c}^j \right),$$

$X_{s,c}^i$ and $X_{t,c}^j$ denote the aligned features of i th and j th training instances from class c in the source and target dataset.

- (6) **for** iter = 1 to 1000 **do**

- (7) Minimize objective function:

$$J = (1/2N) \sum_{i=1}^N \|T_i - h_{W_1, b_1, W_2, b_2}(x_i)\|_2^2 \text{ for both encoders in parallel via stochastic gradient descent.}$$

- (8) **end for**

- (9) Take the activations of aligned-to-generalized encoders as the final generalized features.

Output:

Generalized features $X^* = \{X_k^* \mid X_k^* = (x_{k,1}^*, \dots, x_{k,m_k}^*), X_k^* \in \mathcal{R}^{L \times m_k}, k = 1, \dots, K\}$ across different datasets.

ALGORITHM 1: Framework of Cross-Dataset Feature Alignment and Generalization (CDFAG).

rather than softmax in AGE because SVM classifier could obtain better results, which has been validated in [55, 56]. To perform visible-to-infrared action classification, a SVM classifier with RBF kernel is trained on generalized features extracted from both visible light (source) and infrared (target) datasets and tested on generalized features extracted from unseen instances from infrared (target) dataset, as shown in the bottom of Figure 1. In Section 4, we will show that such classification scheme can effectively classify unseen action data in target dataset. This can be attributed to the successful knowledge transfer from the source domain to the target domain by our proposed CDFAG. To make it clear, the testing procedure of our proposed CDFAG is summarized in Algorithm 2.

4. Experimental Results

In this section, we present our experimental results on the benchmark dataset. We will start with describing the individual datasets, followed by details of our experimental settings.

4.1. Datasets. The InfAR dataset (<https://sites.google.com/site/gaochenqiang/publication/infrared-action-dataset/>) and the XD145 dataset (the dataset will be available at: <https://sites.google.com/site/yangliuxdu/>) are used for the visible-to-infrared action recognition task, where the XD145 dataset is used as the source domain and the InfAR dataset is used as the target domain.

(A) InfAR

The InfAR dataset [20] consists of 600 video sequences captured by infrared thermal imaging

cameras. As shown in Figure 4, fight, handclapping, handshake, hug, jog, jump, punch, push, skip, walk, wave 1 (one-hand wave), and wave 2 (two-hand wave) are included in the dataset, where each action class has 50 video clips and each clip lasts 4 s in average. The frame rate is 25 fps and the resolution is 293×256 . Each video contains one or multiple actions performed by one or several persons. Some of them are interactions between multiple persons, shown in Figure 4.

(B) XD145

We build a visible light action dataset, named XD145, following the approach to construct an action recognition dataset from the visible spectrum [57]. In correspondence with the target domain action categories, both the XD145 and the InfAR dataset have the same action categories, as shown in Figure 5. The XD145 action dataset consists of 600 video sequences captured by visible light cameras and there are 50 video clips for each action class. All actions were performed by 30 different volunteers. Each clip lasts for 5 s in average. The frame rate is 25 fps and the resolution is 320×240 . As shown in Figure 5, the background, pose, and viewpoint variations are considered when constructing the dataset in order to make our dataset more representative for real-world scenarios.

Figure 6 illustrates sample actions from the InfAR and the XD145 datasets. As can be seen in the figure, these action videos are captured in two different light spectra and they

Input:

Raw features of testing samples in target dataset $X_{\text{target,test}} = \{\mathbf{x}_1, \dots, \mathbf{x}_{N_{\text{test}}}\}$, $X_{\text{target,test}} \in \mathcal{R}^{d \times N_{\text{test}}}$, the number of testing samples in target dataset N_{test} , dimension of raw features d , dimension of common latent subspace n , trade-off parameter $\mu = 0.1$, learned projection function for target dataset $\mathbf{f}_{\text{target}}$, trained aligned-to-generalized encoder for target dataset f_1^{target} and f_2^{target} , SVM classifier trained by samples from both source and target dataset.

Feature alignment:

- (1) Map the raw features in target datasets to the Hilbert space:

$$\Phi_{\text{target,test}} = [\phi(\mathbf{x}_1), \dots, \phi(\mathbf{x}_{N_{\text{test}}})]^\top.$$

- (2) Apply $\mathbf{f}_{\text{target}}$ to map the raw features in target datasets to the new n dimensional common latent space to obtain aligned features:

$$\mathbf{X}_{\text{target,aligned}} = \mathbf{f}_{\text{target}}^\top \Phi^i = \alpha^\top \Phi_{\text{target,test}}^\top \Phi^i = \alpha^\top \mathbf{K}_{\text{target,test}}^i, \quad i = 1, \dots, N_{\text{test}}.$$

Feature generalization:

- (3) Input the aligned features $\mathbf{X}_{\text{target,aligned}}$ into the trained aligned-to-generalized encoder and obtained the generalized features $\mathbf{X}_{\text{target,generalized}}$ at the output layer:

$$\mathbf{X}_{\text{target,generalized}} = f_2^{\text{target}}(f_1^{\text{target}}(\mathbf{X}_{\text{target,aligned}}))$$

Classification:

- (4) Adopt the trained SVM classifier to predicts the class labels of testing samples in target dataset using generalized features $\mathbf{X}_{\text{target,generalized}}$.

Output:

Predicted labels of testing samples in the target dataset.

ALGORITHM 2: Testing algorithm of CDFAG.



FIGURE 4: Example images from video sequences in the InfAR dataset.

exhibit significantly great intraclass variance and modality gap.

4.2. Experimental Settings. In all experiments, each dataset is randomly split into training and testing sets. For evaluation, the average precision (AP) is used, which is the average of

recognition precisions of all actions. For each evaluation, we repeat the experiments with the same setting 5 times and report the average accuracy. In KEMA, we use the RBF kernels with the bandwidth fixed as half of the median distance between the samples of the specific video (labeled and unlabeled). By doing so, we allow different kernels in

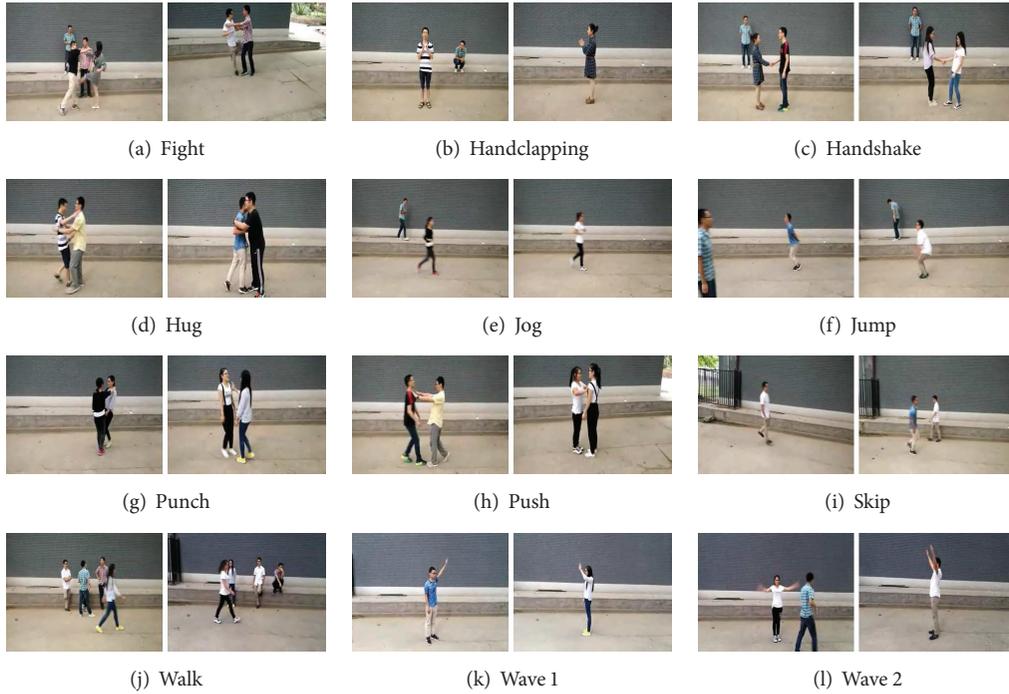


FIGURE 5: 12 actions of the newly constructed visible light action dataset XD145.



FIGURE 6: Comparison of visible and infrared actions. The left of each subfigure is the visible image from video sequences in the XD145 dataset, while the right is the infrared image from video sequences in the InfAR dataset.

each domain, thus tailoring the similarity function to the data structure observed [15]. The trade-off parameter μ is set as 0.1 according to the result of experimental analysis in Section 4.4.3. To build the graph Laplacians, we use a k -NN graph with $k = 10$. We validate the optimal dimension

n of common latent space as well as the optimal C and γ parameters in the SVM classifier. Since the RBF kernel is adopted in KEMA, the LibSVM [58] is used to train classifiers in our experiments with RBF kernel as well. The optimal parameters C and γ are determined by 5-fold

TABLE 1: The average recognition precision (%) of iDTs feature in XD145 and InfAR datasets. train_num and test_num denote the number of samples in training and testing sets, respectively.

Settings	Accuracies in XD145 (%)	Accuracies in InfAR (%)
train_num = 25, test_num = 20	90.58 \pm 2.14	69.25 \pm 2.71
train_num = 20, test_num = 20	90.00 \pm 2.19	66.92 \pm 1.37
train_num = 15, test_num = 20	86.83 \pm 1.78	63.92 \pm 2.24
train_num = 10, test_num = 20	83.58 \pm 2.74	40.33 \pm 7.88
train_num = 5, test_num = 20	77.25 \pm 5.29	30.25 \pm 5.41

cross-validation. When performing stochastic gradient descent for encoder training, we set learning rate $\eta = 0.1$ and momentum $\mu = 0.9$ (see (14)). Each encoder is trained for about 1000 iterations. All experiments are conducted with MATLAB R2016b on a 64-bit Windows 10 PC with 4-core 3.60 GHz Intel i7 CPU and 16 GB of memory.

4.3. Action Recognition Results with Raw Features. We evaluate the original feature in our newly constructed visible light action dataset XD145 and infrared action recognition dataset InfAR, respectively, which is constructed as baseline in this paper. For each evaluation, we repeat the experiments with the same settings 5 times, where, for each time, the numbers of 25/20/15/10/5 samples out of 50 samples for each action category are randomly selected as the training set and 20 samples out of the remaining samples are used as the testing set. Then, the averages are employed as the final result, as shown in Table 1.

From Table 1, we can observe that the recognition accuracies in both datasets are growing with the number of training samples increasing. This is basically in accordance with the traditional action recognition intuition that a good action recognition result can be achieved with adequate amount of labeled training samples and discriminative features. Although the same feature is adopted in both datasets, the accuracies in InfAR are much lower than that in XD145 even with the same experimental setting. This result may be due to the fact that the videos in these two datasets are captured by different sensors and they exhibit variant appearance and motion information for the same action class, as shown in Figure 6. Since the iDTs feature is good at appearance and motion description in visible light action videos [48], its effectiveness and strength may not be revealed in infrared videos. Therefore, utilizing existing visible light action data as an aide for enhancing infrared action recognition system is urgently needed.

4.4. Visible-to-Infrared Action Recognition Results. In this section, we evaluate our proposed method on visible-to-infrared action recognition. Our experiments are mainly divided into four parts. Firstly, a classifier trained on instances from both source and target datasets without feature alignment and generalization is utilized to predict actions in target dataset. Secondly, the classification results with aligned features obtained by KEMA are also provided. Then, we evaluate our proposed CDFAG in visible-to-infrared action recognition. Lastly, we compare our proposed CDFAG with several state-of-the-art methods.

4.4.1. Visible-to-Infrared Action Recognition. To find out whether modality gap is existent between the source and target datasets, we first train a classifier using the samples from both source and target datasets without feature alignment and generalization to predict actions in target dataset. We call this method No Adaptation (NA). In addition, the classification results with aligned features obtained by KEMA are provided to show the effectiveness of feature alignment. For each evaluation we repeat the experiments with the same parameter settings 5 times, where, for each time, the numbers of 45/40/30/25/20/15/10/5 samples for each action category in source dataset combined with the number of 25/20/15/10/5 samples for each action category in target dataset are randomly selected as the training set, in order to validate the impact of the number of training samples on recognition accuracy. Then, 20 samples for each action category in target dataset are used as the testing set. The averages are employed as the final result, as shown in Table 2.

We can see that, in Table 2, the best accuracy of the NA in each column (marked bold) occurs when the number of training samples in source dataset (S.train) is relatively small. Comparing the results of infrared action recognition in Table 1 and the NA in Table 2, we can discover that the No Adaptation (NA) method results in better performance than the baseline method only when fewer samples are used for training ($T_{train} = 10$ and $T_{train} = 5$), which demonstrates that directly transferring the knowledge from the source domain to target domain without considering their divergence can cause significant performance degeneration especially when the number of source domain samples is large. When the number of source and target training samples is small, the performance gets slightly better because of the complementary information between the source and target training samples. However, with the number of source and target training samples increasing, the modality gap between datasets begins dominating the recognition accuracy; then the performance degeneration becomes more serious. Therefore, it is necessary to reduce the modality gap before directly using them together.

Then, we evaluate our proposed CDFAG in visible-to-infrared action recognition. At feature alignment stage, we use the labeled and unlabeled samples to extract KEMA projections and then project all videos in the latent space to obtain aligned samples. We set the dimension of features in common latent space to be 100. We experiment with various feature dimensions and report the results in Section 4.4.3. For videos from the same action class in source dataset XD145, we randomly choose the number of 45/40/35/30/25/20/15/10/5

TABLE 2: The average recognition precision (%) of No Adaptation (NA), KEMA, and the proposed CDFAG in target dataset InfAR. S_{train} and T_{train} denote the number of training samples in source and target datasets, respectively. Best accuracies of each column for each method are marked in italics.

Settings	$T_{train} = 25$	$T_{train} = 20$	$T_{train} = 15$	$T_{train} = 10$	$T_{train} = 5$
$S_{train} = 45$					
NA	59.67 ± 2.22	56.58 ± 1.99	52.58 ± 1.99	41.33 ± 2.56	32.08 ± 2.59
KEMA	66.08 ± 2.56	64.00 ± 3.57	60.25 ± 2.74	56.08 ± 3.89	43.08 ± 4.98
CDFAG	72.50 ± 1.98	69.25 ± 3.71	67.66 ± 3.78	62.25 ± 2.84	47.67 ± 2.91
$S_{train} = 40$					
NA	63.25 ± 1.85	54.00 ± 2.68	50.58 ± 2.68	42.75 ± 3.57	31.25 ± 3.99
KEMA	69.67 ± 1.43	66.25 ± 2.52	59.50 ± 2.56	53.58 ± 2.58	44.50 ± 2.07
CDFAG	75.42 ± 3.13	71.33 ± 1.99	67.58 ± 1.73	62.42 ± 2.59	49.33 ± 3.17
$S_{train} = 35$					
NA	60.67 ± 2.48	57.42 ± 4.05	50.00 ± 2.86	45.75 ± 2.23	32.42 ± 4.79
KEMA	66.67 ± 1.86	64.00 ± 3.92	62.67 ± 2.76	56.92 ± 2.74	41.58 ± 4.14
CDFAG	73.25 ± 1.43	69.25 ± 2.19	65.83 ± 2.37	61.33 ± 2.51	50.06 ± 4.60
$S_{train} = 30$					
NA	62.58 ± 3.10	58.00 ± 2.49	52.17 ± 2.75	44.42 ± 2.27	33.17 ± 1.52
KEMA	68.75 ± 1.86	65.42 ± 2.30	57.67 ± 2.20	55.58 ± 4.16	45.75 ± 4.54
CDFAG	73.83 ± 2.19	69.25 ± 2.35	67.25 ± 2.74	60.58 ± 2.77	50.42 ± 3.90
$S_{train} = 25$					
NA	61.67 ± 4.54	55.92 ± 3.52	49.33 ± 2.51	46.08 ± 2.87	32.33 ± 2.82
KEMA	67.75 ± 2.08	64.17 ± 3.00	60.08 ± 1.78	52.50 ± 2.28	44.92 ± 1.94
CDFAG	72.25 ± 2.18	71.58 ± 2.42	68.08 ± 3.87	60.92 ± 2.15	50.75 ± 2.19
$S_{train} = 20$					
NA	64.17 ± 1.14	59.50 ± 3.08	52.42 ± 3.25	46.25 ± 2.80	33.25 ± 4.42
KEMA	69.67 ± 2.80	67.00 ± 3.15	62.58 ± 2.63	52.42 ± 3.43	46.25 ± 2.38
CDFAG	71.50 ± 2.22	70.58 ± 2.61	66.92 ± 4.10	60.92 ± 3.72	47.00 ± 6.08
$S_{train} = 15$					
NA	63.50 ± 3.21	59.75 ± 5.28	<i>54.17 ± 4.31</i>	46.42 ± 2.35	33.83 ± 3.22
KEMA	67.75 ± 3.90	<i>67.08 ± 3.27</i>	56.33 ± 2.45	55.33 ± 1.54	42.25 ± 4.13
CDFAG	70.75 ± 2.09	69.16 ± 1.79	66.00 ± 2.65	59.58 ± 2.86	47.92 ± 4.59
$S_{train} = 10$					
NA	<i>66.50 ± 2.01</i>	<i>59.75 ± 3.14</i>	54.00 ± 3.63	46.75 ± 3.53	33.58 ± 4.11
KEMA	69.17 ± 3.55	66.83 ± 3.64	62.58 ± 2.64	54.00 ± 5.42	38.83 ± 2.97
CDFAG	71.67 ± 2.43	67.42 ± 1.54	64.92 ± 2.09	61.08 ± 3.14	45.92 ± 3.90
$S_{train} = 5$					
NA	66.33 ± 1.87	53.42 ± 11.15	45.92 ± 11.56	33.92 ± 2.87	30.67 ± 6.82
KEMA	65.00 ± 2.04	65.75 ± 4.20	60.17 ± 3.53	56.17 ± 3, 34	35.67 ± 4.25
CDFAG	72.00 ± 2.72	68.58 ± 3.03	65.67 ± 2.77	57.00 ± 2.85	45.08 ± 3.08

videos as the training set while using the number of 5 videos as the unlabeled set. For videos from the same action class in target dataset InfAR, 20 samples out of 50 samples are randomly selected as the testing set; then we randomly choose the number of 25/20/15/10/5 videos as the training set while using the remaining videos as the unlabeled set, since the target dataset usually has more unlabeled samples but less labeled samples than the source dataset in real-world scenarios. The unlabeled samples are utilized to compute the graph Laplacians. At feature generalization stage, all the aligned samples from both of the source and target training sets are used to guide the training of the two aligned-to-generalized encoders. After feature generalization, a SVM

classifier with RBF kernel is trained on all the aligned and generalized labeled training samples from both source and target datasets. The trained SVM is used to predict all the testing videos in target dataset. All evaluation results are listed in Table 2.

Comparing the results of the NA and the KEMA in Table 2, it is obvious that the KEMA achieves higher accuracies than that of the NA under all parameter settings which validates the effectiveness of KEMA in aligning the features across the source and target domains. In addition, it is obvious that the infrared action recognition accuracies of our proposed CDFAG are significantly much higher than that of No Adaptation (NA) and the KEMA under all

TABLE 3: The average recognition precision (%) in InfAR with different trade-off parameter μ .

	$\mu = 0$	$\mu = 0.1$	$\mu = 0.2$	$\mu = 0.3$	$\mu = 0.4$	$\mu = 0.5$	$\mu = 0.6$	$\mu = 0.7$	$\mu = 0.8$	$\mu = 0.9$	$\mu = 1$
$S_{\text{train}} = 45; T_{\text{train}} = 25$	46.39	72.75	71.83	71.17	71.17	69.50	70.58	67.08	66.33	61.58	61.08
$S_{\text{train}} = 40; T_{\text{train}} = 25$	45.83	73.83	70.42	72.42	72.08	68.92	67.17	66.83	66.42	64.83	56.67
$S_{\text{train}} = 35; T_{\text{train}} = 25$	43.33	72.75	71.42	72.08	70.08	70.17	71.25	70.92	65.17	63.08	53.75

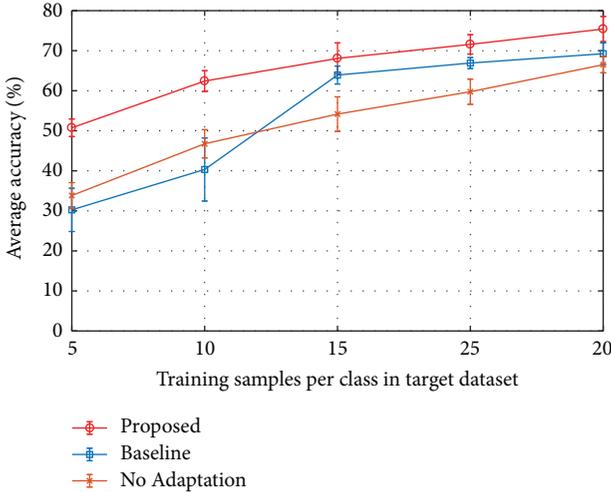


FIGURE 7: Performance comparison of our proposed method with the baseline method and the No Adaptation (NA) method on the InfAR dataset.

parameter settings, which validates the effectiveness of our proposed CDFAG in reducing the modality gap between the source and target datasets by using both feature alignment and generalization. Furthermore, we achieve at least 5% increase in infrared action recognition accuracy over the baseline method in Table 1 (the third column) as a result of utilizing source domain data as an aide for enhancing the present recognition system by a novel feature alignment and generalization method. In order to have a more intuitive comparison, best accuracies under different number of target training samples for each method are plotted in Figure 7. As can be illustrated in Figure 7, our proposed method achieves remarkable performance improvement in infrared action recognition under all parameter settings especially with fewer training samples, which verifies its effectiveness in utilizing auxiliary source domain data under scenario of scarce target training data.

To further explore the infrared action classification performance, two confusion matrices are illustrated in Figure 8. It can be seen that our proposed method achieves higher classification accuracies for nearly all action categories compared with the baseline method. However, there are limited accuracies improvement in two actions—push and punch—since these two actions are easily confused with each other. From Figures 4(g) and 4(h), we can see that punch and push are so similar that it may even be difficult for a human to distinguish them from each other.

4.4.2. Visualization of Aligned and Generalized Features. To verify that, by using our proposed CDFAG, all action data from different datasets are indeed projected into the unique common latent space, we plotted the distribution of the raw features, aligned features, and generalized features of instances from all action classes in source and target datasets, respectively. For illustration purposes, we compared the first 3-dimensional feature space of the raw iDTs features, aligned features, and the learnt generalized features, shown in Figure 9. As can be seen in Figure 9(a), the original features of instances from two datasets are obviously in different clusters with large intraclass diversity and small interclass variance, but they are projected into a single cluster where the instances from the same class are projected to similar locations and the instances from different classes are well-separated, as illustrated in Figure 9(b). Although there is small distinction between Figures 9(b) and 9(c) except that the instances from the same class in Figure 9(c) merge into a more compact cluster, feature generalization indeed maps instances from different datasets to a more compact feature space with relatively small intraclass diversity and large interclass variance, which makes features more generalized and discriminative for visible-to-infrared action recognition.

4.4.3. Analysis of Trade-Off Parameter μ . To evaluate the optimum value range of trade-off parameter μ in (1), we evaluate the performance of our proposed CDFAG with different values of μ . Specifically, we use $\mu = \{0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1\}$ in InfAR dataset when (1) $S_{\text{train}} = 45$ and $T_{\text{train}} = 25$, (2) $S_{\text{train}} = 40$ and $T_{\text{train}} = 25$, and (3) $S_{\text{train}} = 35$ and $T_{\text{train}} = 25$. The results of other S_{train} and T_{train} settings are not evaluated because other settings show similar results to these three settings. For each evaluation, we repeat the experiments with the same settings 5 times and report the average accuracies.

The experimental results are given in Table 3. As shown in Table 3, the average accuracies of these three settings are better when trade-off parameter μ is small, which indicates that a good performance can be achieved when we attach more importance to class similarity minimization term than topology preservation in KEMA procedure. When we treat both terms equally ($\mu = 0.5$), the performance is also unsatisfactory. These attribute to the fact that the modality difference between infrared and visible light domain is so huge that the topology between them also varies a lot. Therefore, more importance should be attached to class similarity term to achieve good performance. However, when μ is set as 0, the overall performance drops dramatically. This shows that the topology preservation plays an important role in KEMA and

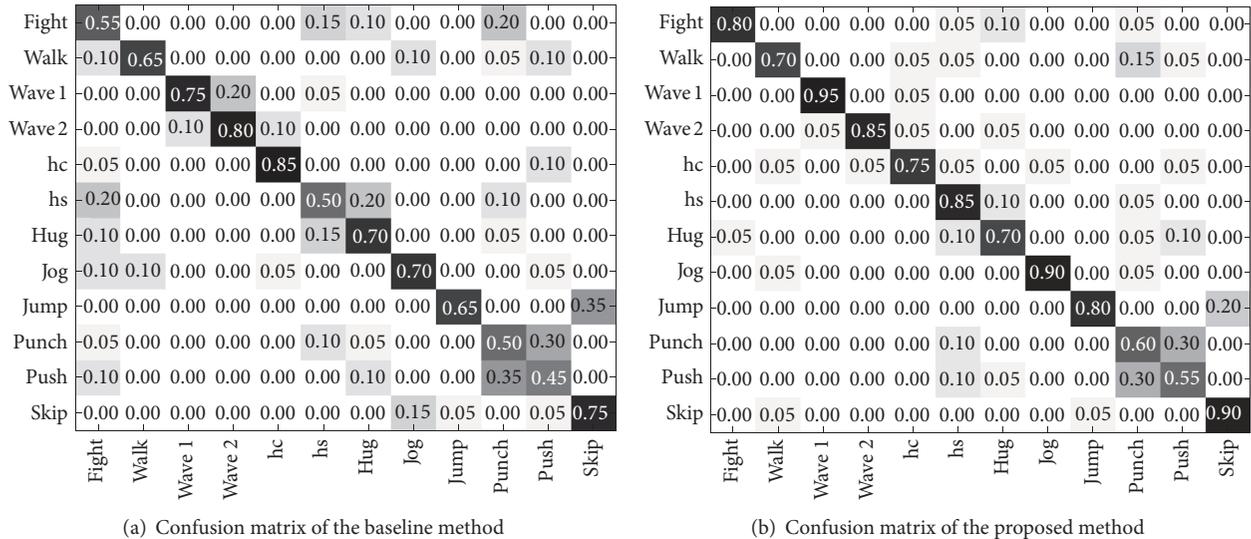


FIGURE 8: The comparison of the confusion matrices between the baseline method and the proposed method in InfAR dataset, where the left is from the baseline method, while the right is from our proposed CDFAG. Note that “hc” denotes “handclapping” and “hs” denotes “handshaking,” respectively.

cannot be neglected. When μ is set as 1, the performance also drops but with less accuracy gap than the case when $\mu = 0$. This means that topology preservation contributes more to the overall performance than class similarity minimization although both of them are necessary in KEMA.

More intuitive analysis is plotted in Figure 10. From Figure 10 we can see that a good result can be achieved when $\mu < 0.5$ and the optimum value of μ is 0.1. Therefore, we set $\mu = 0.1$ in the experiments. In addition, the performance is relatively stable with small accuracy gap when $\mu < 0.5$, which shows that our algorithm is insensitive to the trade-off parameter μ when $\mu < 0.5$.

4.4.4. Feature Dimension in Common Latent Space. We experiment with various feature dimensions n in common latent space to study how the feature dimension influences the classification accuracy. As the hidden layer size H in aligned-to-generalized encoders is empirically set to be equal to the input layer size L , the hidden layer size H is directly determined by the value of the feature dimension n in latent space. As illustrated in Figure 11, our proposed method arrives at its best accuracy when feature dimension $n = 100$; then classification accuracy tends to decrease as the feature dimension increases. This can be explained by the decreased discriminability of the feature representations in common latent space as its dimension increases.

4.4.5. Computation Time. We evaluate the computation time of our proposed method and report the results in Table 4. All experiments are conducted on our lab PC and developed by MATLAB. The reported times are averaging running time over all the S_{train} values when the T_{train} remains unchanged, and the computation time for iDTs features and LLC encoding is not included. The reported time includes

feature alignment, feature generalization, classifier training, cross-validation and classification. As can be seen from Table 4, our proposed method can perform feature alignment, feature generalization, classifier training, and classification very efficiently. The longest computation time is just nearly 9 minutes when $n = 500$ and $T_{train} = 25$. We attribute this efficient execution to the PCA dimension reduction of raw features, fast feature alignment by introducing the kernel trick in KEMA, and the shallow single-hidden-layer neural network architecture for feature generalization.

4.4.6. Comparison with State-of-the-Art Methods. We compare our proposed CDFAG method with the following state-of-the-art methods:

- (i) Domain adaptation based methods: KEMA [15], SSMA [16], and DA [17].
- (ii) Transfer learning based methods: WSCDDL [18] and Dual [19].
- (iii) Deep learning based methods: two-stream CNNs [20].

Actually, we focus on the transferable features generation in this paper and only use the features generated by these comparison methods including KEMA, SSMA, DA, WSCDDL, and Dual. And then we use SVM as the classifier to recognize the actions. In order to compare with these domain adaptation and transfer learning methods, we use the same experimental setting as our proposed CDFAG (Section 4.4.1) in these related works. Results are reported in Table 5. The results in boldface in each column shows that the proposed CDFAG is the most competitive one compared with other state-of-the-art methods. For instance, the average accuracy of the proposed method CDFAG brings about 20.50%, 22.09%, 4.16%, 4.66%, and 6.17% improvements over the

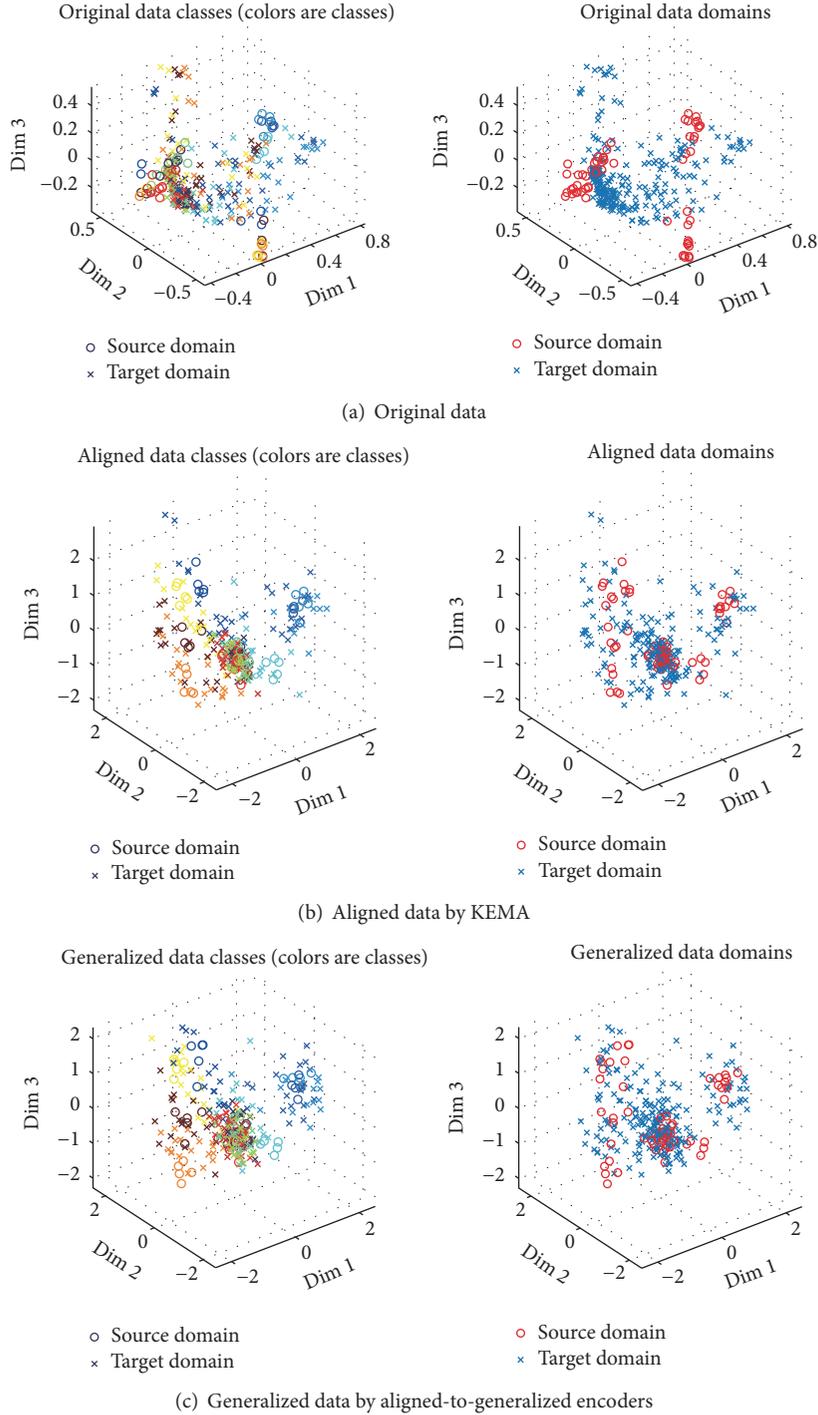


FIGURE 9: Examples of the three first dimensions of the original, aligned, and generalized data space. The left panel shows data distributions of two domains per class while the right panel shows the data distributions of all classes per domain.

baseline method for five different T_{train} values, respectively. This validates the effectiveness of the CDFAG in improving the overall infrared action recognition accuracy with the aid of the visible light data from source domain. Compared with domain adaptation methods, the average accuracy of the CDFAG is about 6.00%, 6.59%, and 11.75% higher than that of KEMA [15], SSMA [16], and DA [17] when $T_{\text{train}} = 25$,

respectively. For transfer learning methods, our proposed method also brings about 8.00% and 17.17% improvements over WSCDDL [18] and Dual [19] when $T_{\text{train}} = 25$, respectively. More comparisons are plotted in Figure 12.

The results in Figure 11 show that the proposed CDFAG performs much better than other state-of-the-art methods on visible-to-infrared action recognition task. As can be

TABLE 4: Average computation time of our proposed method CDFAG. T_{train} denotes the number of training samples in the target dataset. n denotes the feature dimension in the common latent space.

	$T_{train} = 5$	$T_{train} = 10$	$T_{train} = 15$	$T_{train} = 20$	$T_{train} = 25$
$n = 50$	46.55 s	58.60 s	73.58 s	87.33 s	106.91 s
$n = 100$	64.22 s	87.85 s	99.54 s	118.44 s	147.49 s
$n = 150$	80.94 s	103.12 s	128.87 s	156.53 s	187.02 s
$n = 200$	98.57 s	126.63 s	159.42 s	200.74 s	229.20 s
$n = 250$	132.59 s	153.89 s	185.90 s	232.15 s	241.18 s
$n = 300$	161.46 s	202.49 s	248.43 s	291.85 s	304.47 s
$n = 350$	181.34 s	223.77 s	274.69 s	345.99 s	423.55 s
$n = 400$	183.73 s	236.03 s	298.81 s	408.99 s	478.81 s
$n = 450$	187.53 s	257.06 s	328.63 s	415.07 s	511.29 s
$n = 500$	212.11 s	266.83 s	338.42 s	415.50 s	546.41 s

TABLE 5: Average recognition precision (%) comparison of the proposed method CDFAG with state-of-the-art methods. For each method, the best accuracies of various T_{train} values are shown. The best accuracies of each T_{train} are marked in italics.

Method	$T_{train} = 5$	$T_{train} = 10$	$T_{train} = 15$	$T_{train} = 20$	$T_{train} = 25$
KEMA [15]	46.92 ± 3.26	55.75 ± 4.92	60.92 ± 2.97	63.92 ± 3.37	69.42 ± 3.44
SSMA [16]	44.33 ± 2.90	58.58 ± 1.83	63.00 ± 2.38	64.92 ± 2.86	68.83 ± 1.12
DA [17]	41.67 ± 2.52	47.83 ± 3.39	54.17 ± 1.41	55.17 ± 1.52	63.67 ± 1.87
WSCDDL [18]	45.92 ± 1.26	56.08 ± 4.23	62.58 ± 2.52	66.08 ± 2.03	67.42 ± 2.42
Dual [19]	35.75 ± 3.23	41.83 ± 5.48	52.25 ± 2.54	58.92 ± 2.22	58.25 ± 4.32
No Adaptation	33.83 ± 3.22	46.75 ± 3.53	54.17 ± 4.31	59.75 ± 3.14	66.50 ± 2.01
Baseline	30.25 ± 5.41	40.33 ± 7.88	63.92 ± 2.24	66.92 ± 1.37	69.25 ± 2.71
CDFAG	<i>50.75 ± 2.19</i>	<i>62.42 ± 2.59</i>	<i>68.08 ± 3.87</i>	<i>71.58 ± 2.42</i>	<i>75.42 ± 3.13</i>

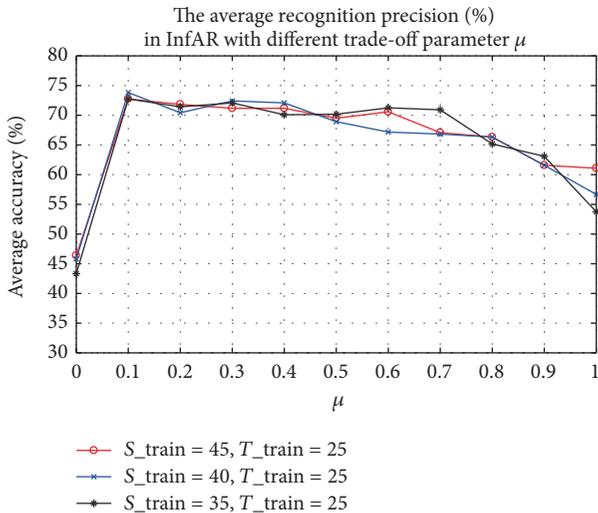


FIGURE 10: Trade-off parameter μ and the average recognition precision (%) in InfAR.

seen from Figure 11, the performance of Dual [19] is not good especially with fewer target training samples because it needs adequate training samples to be well-trained. On the one hand, other state-of-the-art methods can achieve higher accuracies than that of the baseline method with fewer target training samples, while, with more target training samples, modality gap begins dominating the recognition

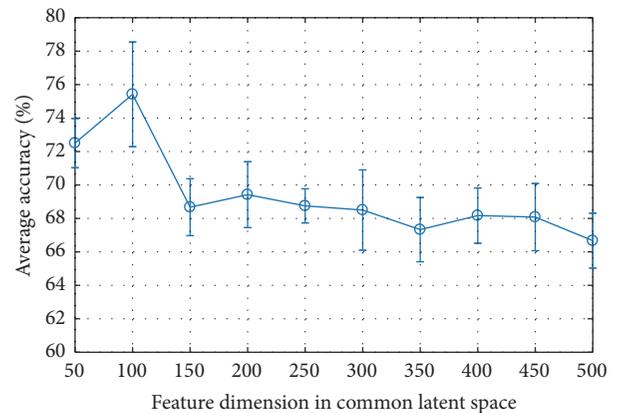


FIGURE 11: Feature dimension n in latent space and classification accuracy (%).

accuracies and their performances become inferior to that of the baseline. On the other hand, our proposed method CDFAG can perform well whenever the number of training samples is small or large, which validates the effectiveness of our proposed method CDFAG in terms of reducing the modality gap across the source and target datasets. The key difference between our proposed CDFAG and the other methods is that our method takes both feature alignment and feature generalization into consideration. Thus, a latent common feature space with low intraclass diversity and high

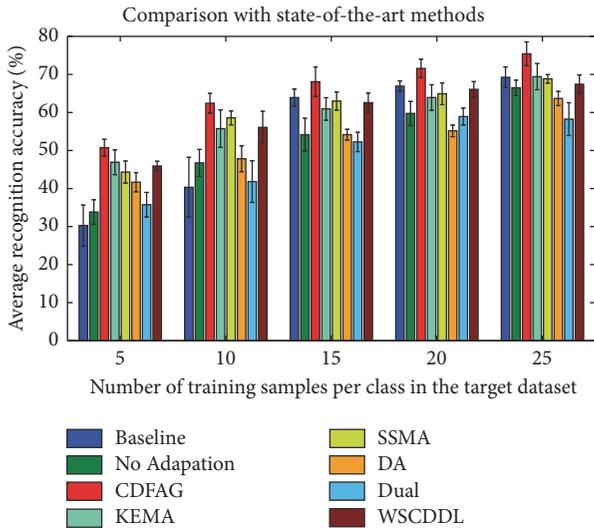


FIGURE 12: Comparison between our results and related works.

interclass variance can be learnt. The other methods focus on only either of them (feature alignment or generalization) without explicit knowledge transfer and effective modality gap reduction. The strong performance of our proposed CDFAG demonstrates the advantage of transferring knowledge from visible light domain to infrared domain and, more importantly, the efficacy of the proposed cross-dataset feature alignment and generalization (CDFAG) framework.

We also compare the proposed CDFAG with state-of-the-art deep learning based method. Two-stream CNN is adopted in [20] and achieved average precision 76.66% in InfAR dataset, while the best accuracy of our proposed method is 75.42%. It is evident that our proposed CDFAG can achieve comparable infrared action recognition performance to that of the deep learning based method in [20]. In addition, the proposed CDFAG can still achieve good performance efficiently with fewer labeled training samples in target dataset while the deep learning based method in [20] is time-consuming and needs a large amount of training instances or pretrained models to perform well. Therefore, our proposed CDFAG may be a good visible-to-infrared action recognition framework that makes a good balance between classification accuracy and time efficiency.

5. Conclusion and Future Work

In this paper, we propose a novel Cross-Dataset Feature Alignment and Generalization (CDFAG) framework for visible-to-infrared action recognition. The proposed CDFAG is essentially a feature extractor that finds projections from all the source and target domains into a common latent feature space where features of all instances are semantically aligned and generalized with low intraclass diversity and high interclass variance. Promising results are achieved on visible-to-infrared action recognition and cross-dataset recognition tasks, where auxiliary source domain knowledge is effectively

transferred to target domain. Compared with several state-of-the-art transfer learning or domain adaptation based methods, our proposed CDFAG offers a more flexible framework and achieves the best performance ($AP = 75.42\%$) in infrared action recognition. In addition, our proposed CDFAG also achieves comparable infrared action recognition performance to deep learning based method. In the future, we will adapt the CDFAG method to other cross-domain action recognition tasks, such as cross-view, cross-dataset, and image-to-video action recognition. Another interesting direction is to modify existing visible light action recognition methods for infrared datasets.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (no. 61502364), the China Postdoctoral Science Foundation funded project (Grant no. 154906), and the Fundamental Research Funds for the Central Universities (Grant no. 3102016ZY022).

References

- [1] X. Wang, "Intelligent multi-camera video surveillance: A review," *Pattern Recognition Letters*, vol. 34, no. 1, pp. 3–19, 2013.
- [2] M. Gori, M. Lippi, M. Maggini, and S. Melacci, "Semantic video labeling by developmental visual agents," *Computer Vision and Image Understanding*, vol. 146, pp. 9–26, 2016.
- [3] W. Hu, D. Xie, Z. Fu, W. Zeng, and S. Maybank, "Semantic-based surveillance video retrieval," *IEEE Transactions on Image Processing*, vol. 16, no. 4, pp. 1168–1181, 2007.
- [4] W. Zhang, M. L. Smith, L. N. Smith, and A. Farooq, "Gender and gaze gesture recognition for human-computer interaction," *Computer Vision and Image Understanding*, vol. 149, pp. 32–50, 2016.
- [5] W. Chen, T. Lao, J. Xia et al., "GameFlow: Narrative Visualization of NBA Basketball Games," *IEEE Transactions on Multimedia*, vol. 18, no. 11, pp. 2247–2256, 2016.
- [6] J. K. Aggarwal and M. S. Ryoo, "Human activity analysis: a review," *ACM Computing Surveys*, vol. 43, no. 3, article 16, 2011.
- [7] I. Laptev, "On space-time interest points," *International Journal of Computer Vision*, vol. 64, no. 2-3, pp. 107–123, 2005.
- [8] H. Wang, A. Kläser, C. Schmid, and C.-L. Liu, "Action recognition by dense trajectories," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '11)*, pp. 3169–3176, June 2011.
- [9] H. Bilen, B. Fernando, E. Gavves, A. Vedaldi, and S. Gould, "Dynamic image networks for action recognition," in *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016*, pp. 3034–3042, usa, July 2016.
- [10] C. Schüldt, I. Laptev, and B. Caputo, "Recognizing human actions: a local SVM approach," in *Proceedings of the 17th International Conference on Pattern Recognition (ICPR '04)*, pp. 32–36, August 2004.

- [11] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre, "HMDB: a large video database for human motion recognition," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV '11)*, pp. 2556–2563, November 2011.
- [12] K. Soomro, A. R. Zamir, and M. Shah, "UCF101: A dataset of 101 human actions classes from videos in the wild, CoRR abs/1212.0402".
- [13] C. Gao, Y. Du, J. Liu, L. Yang, and D. Meng, "A new dataset and evaluation for infrared action recognition," *Communications in Computer and Information Science*, vol. 547, pp. 302–312, 2015.
- [14] J. Han and B. Bhanu, "Fusion of color and infrared video for moving human detection," *Pattern Recognition*, vol. 40, no. 6, pp. 1771–1784, 2007.
- [15] D. Tuia and G. Camps-Valls, "Kernel manifold alignment for domain adaptation," *PLoS ONE*, vol. 11, no. 2, Article ID e0148655, 2016.
- [16] C. Wang and S. Mahadevan, "Heterogeneous domain adaptation using manifold alignment," in *Proceedings of the 22nd International Joint Conference on Artificial Intelligence*, pp. 1541–1546, July 2011.
- [17] B. Fernando, A. Habrard, M. Sebban, and T. Tuytelaars, "Unsupervised visual domain adaptation using subspace alignment," in *Proceedings of the 2013 14th IEEE International Conference on Computer Vision, ICCV 2013*, pp. 2960–2967, Australia, December 2013.
- [18] F. Zhu and L. Shao, "Weakly-supervised cross-domain dictionary learning for visual recognition," *International Journal of Computer Vision*, vol. 109, no. 1-2, pp. 42–59, 2014.
- [19] T. Xu, F. Zhu, E. K. Wong, and Y. Fang, "Dual many-to-one-encoder-based transfer learning for cross-dataset human action recognition," *Image and Vision Computing*, vol. 55, pp. 127–137, 2016.
- [20] C. Gao, Y. Du, J. Liu et al., "InfAR dataset: Infrared action recognition at different times," *Neurocomputing*, vol. 212, pp. 36–47, 2016.
- [21] X. Yao, J. Han, D. Zhang, and F. Nie, "Revisiting co-saliency detection: a novel approach based on two-stage multi-view spectral rotation co-clustering," *IEEE Transactions on Image Processing*, vol. 26, no. 7, pp. 3196–3209, 2017.
- [22] G. Cheng, Z. Li, X. Yao, L. Guo, and Z. Wei, "Remote Sensing Image Scene Classification Using Bag of Convolutional Features," *IEEE Geoscience and Remote Sensing Letters*, vol. 14, no. 10, pp. 1735–1739, 2017.
- [23] L. Yang, X. Cai, S. Pan, H. Dai, and D. Mu, "Multi-document summarization based on sentence cluster using non-negative matrix factorization," *Journal of Intelligent & Fuzzy Systems: Applications in Engineering and Technology*, vol. 33, no. 3, pp. 1867–1879, 2017.
- [24] C. Yao, J. Han, F. Nie, F. Xiao, and X. Li, "Local regression and global information-embedded dimension reduction, to be published," *IEEE Transactions on Neural Networks and Learning Systems*, 2017.
- [25] C. Yao, Y.-F. Liu, B. Jiang, J. Han, and J. Han, "LLE score: a new filter-based unsupervised feature selection method based on nonlinear manifold embedding and its application to image recognition," *IEEE Transactions on Image Processing*, vol. 26, no. 11, pp. 5257–5269, 2017.
- [26] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 10, pp. 1345–1359, 2010.
- [27] V. M. Patel, R. Gopalan, R. Li, and R. Chellappa, "Visual Domain Adaptation: A survey of recent advances," *IEEE Signal Processing Magazine*, vol. 32, no. 3, pp. 53–69, 2015.
- [28] L. Zhang, W. Zuo, and D. Zhang, "LSDT: latent sparse domain transfer learning for visual adaptation," *IEEE Transactions on Image Processing*, vol. 25, no. 3, pp. 1177–1191, 2016.
- [29] L. Zhang, J. Yang, and D. Zhang, "Domain class consistency based transfer learning for image classification across domains," *Information Sciences*, vol. 418–419, pp. 242–257, 2017.
- [30] L. Zhang and D. Zhang, "Robust visual knowledge transfer via extreme learning machine-based domain adaptation," *IEEE Transactions on Image Processing*, vol. 25, no. 10, pp. 4959–4973, 2016.
- [31] J. Ham, D. D. Lee, and L. K. Saul, "Semisupervised alignment of manifolds," in *Proceedings of the 10th International Workshop on Artificial Intelligence and Statistics, AISTATS 2005*, pp. 120–127, brb, January 2005.
- [32] D. Tuia, D. Marcos, and G. Camps-Valls, "Multi-temporal and multi-source remote sensing image classification by nonlinear relative normalization," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 120, pp. 1–12, 2016.
- [33] D. H. Hu, V. W. Zheng, and Q. Yang, "Cross-domain activity recognition via transfer learning," *Pervasive and Mobile Computing*, vol. 7, no. 3, pp. 344–358, 2011.
- [34] A. Farhadi and M. K. Tabrizi, "Learning to Recognize Activities from the Wrong View Point," in *Computer Vision – ECCV 2008*, vol. 5302 of *Lecture Notes in Computer Science*, pp. 154–166, Springer Berlin Heidelberg, Berlin, Heidelberg, 2008.
- [35] J. Zheng, Z. Jiang, P. Jonathon Phillips, and R. Chellappa, "Cross-view action recognition via a transferable dictionary pair," in *Proceedings of the 2012 23rd British Machine Vision Conference, BMVC 2012*, gbr, September 2012.
- [36] Z. Zhang, C. Wang, B. Xiao, W. Zhou, S. Liu, and C. Shi, "Cross-view action recognition via a continuous virtual path," in *Proceedings of the 26th IEEE Conference on Computer Vision and Pattern Recognition (CVPR '13)*, pp. 2690–2697, June 2013.
- [37] X. Wu, H. Wang, C. Liu, and Y. Jia, "Cross-view action recognition over heterogeneous feature spaces," *IEEE Transactions on Image Processing*, vol. 24, no. 11, pp. 4096–4108, 2015.
- [38] W. Sui, X. Wu, Y. Feng, and Y. Jia, "Heterogeneous discriminant analysis for cross-view action recognition," *Neurocomputing*, vol. 191, pp. 286–295, 2016.
- [39] C. Zu and D. Zhang, "Canonical sparse cross-view correlation analysis," *Neurocomputing*, vol. 191, pp. 263–272, 2016.
- [40] W. Bian, D. Tao, and Y. Rui, "Cross-domain human action recognition," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 42, no. 2, pp. 298–307, 2012.
- [41] N. C. Tang, Y.-Y. Lin, J.-H. Hua, S.-E. Wei, M.-F. Weng, and H. M. Liao, "Robust action recognition via borrowing information across video modalities," *IEEE Transactions on Image Processing*, vol. 24, no. 2, pp. 709–723, 2015.
- [42] F. Liu, X. Xu, S. Qiu, C. Qing, and D. Tao, "Simple to complex transfer learning for action recognition," *IEEE Transactions on Image Processing*, vol. 25, no. 2, pp. 949–960, 2016.
- [43] M. Kan, S. Shan, and X. Chen, "Bi-shifting auto-encoder for unsupervised domain adaptation," in *Proceedings of the 15th IEEE International Conference on Computer Vision, ICCV 2015*, pp. 3846–3854, Chile, December 2015.
- [44] J. Han and B. Bhanu, "Human Activity Recognition in Thermal Infrared Imagery," in *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Workshops*, p. 17, San Diego, CA, USA.

- [45] H. Eum, J. Lee, C. Yoon, and M. Park, "Human action recognition for night vision using temporal templates with infrared thermal camera," in *Proceedings of the 2013 10th International Conference on Ubiquitous Robots and Ambient Intelligence, URAI 2013*, pp. 617–621, Republic of Korea, November 2013.
- [46] Y. Zhu and G. Guo, "A study on visible to infrared action recognition," *IEEE Signal Processing Letters*, vol. 20, no. 9, pp. 897–900, 2013.
- [47] J. Yang, R. Yan, and A. G. Hauptmann, "Cross-domain video concept detection using adaptive SVMs," in *Proceedings of the 15th ACM International Conference on Multimedia (MM '07)*, pp. 188–197, September 2007.
- [48] H. Wang and C. Schmid, "Action recognition with improved trajectories," in *Proceedings of the 14th IEEE International Conference on Computer Vision (ICCV '13)*, pp. 3551–3558, Sydney, Australia, December 2013.
- [49] J. Wang, J. Yang, K. Yu, F. Lv, T. Huang, and Y. Gong, "Locality-constrained linear coding for image classification," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '10)*, pp. 3360–3367, IEEE, June 2010.
- [50] L. I. Smith, "A tutorial on principal components analysis," *Information Fusion*, vol. 51, no. 3, pp. 219–226, 2002.
- [51] B. L. V. D. Waerden, *The Mathematical Gazette*, Prentice-Hall Internations, Inc, 1971.
- [52] S. Yan, D. Xu, B. Zhang, H. Zhang, Q. Yang, and S. Lin, "Graph embedding and extensions: a general framework for dimensionality reduction," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 1, pp. 40–51, 2007.
- [53] Y. Fang, J. Xie, G. Dai et al., "3D deep shape descriptor," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015*, pp. 2319–2328, USA, June 2015.
- [54] A. Coates, H. Lee, and A. Y. Ng, "An analysis of single-layer networks in unsupervised feature learning," *Journal of Machine Learning Research*, vol. 15, pp. 215–223, 2011.
- [55] K. Nogueira, O. A. B. Penatti, and J. A. dos Santos, "Towards better exploiting convolutional neural networks for remote sensing scene classification," *Pattern Recognition*, vol. 61, pp. 539–556, 2017.
- [56] G. Cheng, C. Yang, X. Yao, L. Guo, and J. Han, "When Deep Learning Meets Metric Learning: Remote Sensing Image Scene Classification via Learning Discriminative CNNs," *IEEE Transactions on Geoscience and Remote Sensing*, pp. 1–11.
- [57] J. M. Chaquet, E. J. Carmona, and A. Fernández-Caballero, "A survey of video datasets for human action and activity recognition," *Computer Vision and Image Understanding*, vol. 117, no. 6, pp. 633–659, 2013.
- [58] C. Chang and C. Lin, "LIBSVM: a Library for support vector machines," *ACM Transactions on Intelligent Systems and Technology*, vol. 2, no. 3, article 27, 2011.

Research Article

Efficient Privacy-Preserving Protocol for k -NN Search over Encrypted Data in Location-Based Service

Huijuan Lian,¹ Weidong Qiu,¹ Di Yan,² Zheng Huang,¹ and Jie Guo¹

¹*School of Cyber Security, Shanghai Jiao Tong University, Shanghai 200240, China*

²*Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai, China*

Correspondence should be addressed to Weidong Qiu; qiuwd@sjtu.edu.cn

Received 1 September 2017; Revised 13 November 2017; Accepted 20 November 2017; Published 20 December 2017

Academic Editor: Jia Wu

Copyright © 2017 Huijuan Lian et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

With the development of mobile communication technology, location-based services (LBS) are booming prosperously. Meanwhile privacy protection has become the main obstacle for the further development of LBS. The k -nearest neighbor (k -NN) search is one of the most common types of LBS. In this paper, we propose an efficient private circular query protocol (EPCQP) with high accuracy rate and low computation and communication cost. We adopt the Moore curve to convert two-dimensional spatial data into one-dimensional sequence and encrypt the points of interest (POIs) information with the Brakerski-Gentry-Vaikuntanathan homomorphic encryption scheme for privacy-preserving. The proposed scheme performs the secret circular shift of the encrypted POIs information to hide the location of the user without a trusted third party. To reduce the computation and communication cost, we dynamically divide the table of the POIs information according to the value of k . Experiments show that the proposed scheme provides high accuracy query results while maintaining low computation and communication cost.

1. Introduction

Nowadays, the location-based services are developing rapidly with the wide use of mobile Internet and smart mobile devices. Location-based services use mobile device to learn the current location with the help of built-in positioning devices and get the location information through the mobile network.

In a location-based service, a user obtains the query result by providing accurate locations to the service provider. The user privacy is probably obtained by the adversary through the conjunction of relevant background knowledge and the captured location information. How to utilize the service while protecting the location privacy of the user has become a research topic in location-based services in recent years.

The privacy data of location-based services includes three aspects: identity information, location information, and query content. The privacy of location refers to hiding the accurate location of the user. The privacy of query content refers to hiding the specific description of the request

submitted by the user. When the query content is obtained, the characteristics and the behaviors of the user can be deduced.

In the LBS, the core of privacy-preserving is to cut off the relevance of the identity information, location information and query content. The query contents from the user are relevant to each other in the case of continuous querying. Therefore, it not only enhances the difficulty of privacy protection, but also increases the computation cost.

There are two communication modes in the privacy-preserving query in LBS: one is based on the trusted third party (TTP) and the other is without trusted third party (TTP-free). Although TTP-based solutions [1–7] are able to collect enough information to maximally meet the needs of privacy protection, there are two problems: (1) It is difficult to obtain the TTP that fill the bill. (2) Centralized attack on TTP makes it become the bottleneck in the query scheme. TTP-free-based solutions take advantage of the limited information to help maximize privacy protection. However, there

is a lack of methods which are outstanding in all three aspects of query accuracy, efficiency, and privacy-preserving.

Lien et al. [8] have proposed a private circular query protocol (PCQP) without a TTP, which uses Moore curve and Paillier cryptosystem to implement the protection of the location and query content. The scheme contains a large number of homomorphic additions and multiplications, and thus it requires higher computation and communication cost. Utsunomiya et al. [9] made some improvements on the basis of PCQP and proposed a lightweight private circular query protocol (LPCQP) with divided POI-table to effectively reduce the number of homomorphic additions and multiplications. The dividing of the POI-table is performed only once in the initialization process and the number of subtables observably influences the accuracy of the query. In some extreme cases, the scheme cannot return enough k POIs to the user. In addition, when the number of POIs in a subtable is much larger than k , the homomorphic additions and multiplications bring large number of unnecessary computation cost. LPCQP uses the homomorphic encryption scheme proposed by Smart and Vercauteren to ensure the security. The scheme needs a large size of public key and a large amount of computation.

Considering the advantages and disadvantages of the above two schemes, this paper proposes an efficient private circular query protocol (EPCQP) to mitigate the drawbacks of LPCQP without damaging the security. The proposed scheme utilizes the fully homomorphic encryption scheme to address the problem of secure querying over encrypted data in LBS. To omit the redundant homomorphic additions and multiplications, the proposed scheme dynamically divides the encrypted POI-table according to the query requirement of the user. The data security depends on the Brakerski-Gentry-Vaikuntanathan (BGV) homomorphic encryption scheme [10]. In addition, the user utilizes the circular shift and modulo operation to replace the real location which guarantees the location privacy-preserving.

The proposed scheme has the following advantages.

(1) *Location Privacy and Data Privacy.* The proposed scheme can defend the correlation attack, the background knowledge attack, the offline keyword guessing attack, the inference attack, the man-in-the-middle attack, and the link attack.

(2) *Computation Efficiency.* The computation cost is reduced by 99% or more when k is from 5 to 50 compared with that of PCQP. When k is smaller than 25, the computation cost of EPCQP is significantly lower than that of LPCQP.

(3) *High Accuracy Rate.* The accuracy rate of the proposed scheme is higher than 90% even if k is large for the uniform dataset, and it is higher than 84% when k is large for the real-world dataset.

The remainder of the paper is organized as follows. Section 2 introduces the relevant background knowledge. Section 3 describes the proposed protocol in detail and Section 4 discusses the performance of the proposed scheme based on various experiments. Related work is reviewed in Section 5 and the conclusions are drawn in Section 6.

2. Background

In this section, we review the main techniques which are utilized in the proposed protocol.

2.1. Moore Curve. Space-filling curves [11] represent a class of curves which traverse through all points in a two-dimensional region or more generally an n -dimensional hypercube, without crossing themselves. Hilbert [12] demonstrated the general geometrical generating procedure for constructing an entire class of space-filling curves in 1891. Hilbert curve has the capability of superior clustering and partially retaining the neighboring adjacency of the original data [13, 14]. Figure 1 illustrates the Hilbert curves of different orders. An N -th order Hilbert curve can pass through all cells in a $2^N \times 2^N$ square grids. The number on the corner of each cell denotes an index, called *H-value*, from the set $[0, 2^N - 1]$ when the curve traverses the cells.

Moore curve [15] is a variation of the Hilbert curve with end-point-connected property. Figure 2 illustrates the Moore curves of different orders. The POIs in a two-dimensional region can be stringed into a circular structure. Our scheme adopts Moore curve due to the circularly connected property.

2.2. Homomorphic Encryption. Without the trusted third party, our scheme adopts the homomorphic encryption to protect the data privacy. Due to the property of the homomorphic encryption showed in (1), the server can perform addition or multiplication of plaintexts without decryption. In (1), m_1, m_2, pk , and sk denote the two plaintexts, the public key, and the private key, respectively. The encryption and decryption are denoted as $\epsilon_{pk}(m)$ and $D_{sk}(m)$. Here, $+_c$ and \times_c denote the homomorphic addition and multiplication over ciphertexts.

$$\begin{aligned} D_{sk}(\epsilon_{pk}(m_1) +_c \epsilon_{pk}(m_2)) &= m_1 + m_2 \\ D_{sk}(\epsilon_{pk}(m_1) \times_c \epsilon_{pk}(m_2)) &= m_1 \times m_2. \end{aligned} \quad (1)$$

Homomorphic encryption [16] was proposed by Rivest et al. in 1978. Gentry [17] proposed the first fully homomorphic encryption scheme in 2009, which used bootstrapping to construct a fully homomorphic encryption scheme from a somewhat homomorphic encryption scheme. The scheme can be summarized as follows: FHE = SHE + Bootstrapping, and its security depends on certain worst-case problems over ideal lattices. In order to reduce the computational complexity of the decryption circuit, he designed a lattice-based decryption circuit, and its security depends on the assumed hardness of two problems: sparse subset sum problem (SSSP) and the certain worst-case problems over ideal lattices. Gentry's scheme uses matrix operations and vector modular arithmetic, resulting in discontinuities in computation and fast growth in computation cost. In addition, the size of the ciphertext generated by the corresponding 1-bit plaintext is exponentially increasing; thus the size is becoming too long to be realized by programming. Despite all of these shortcomings, Gentry still makes a great contribution to the study of the fully homomorphic encryption. To improve the

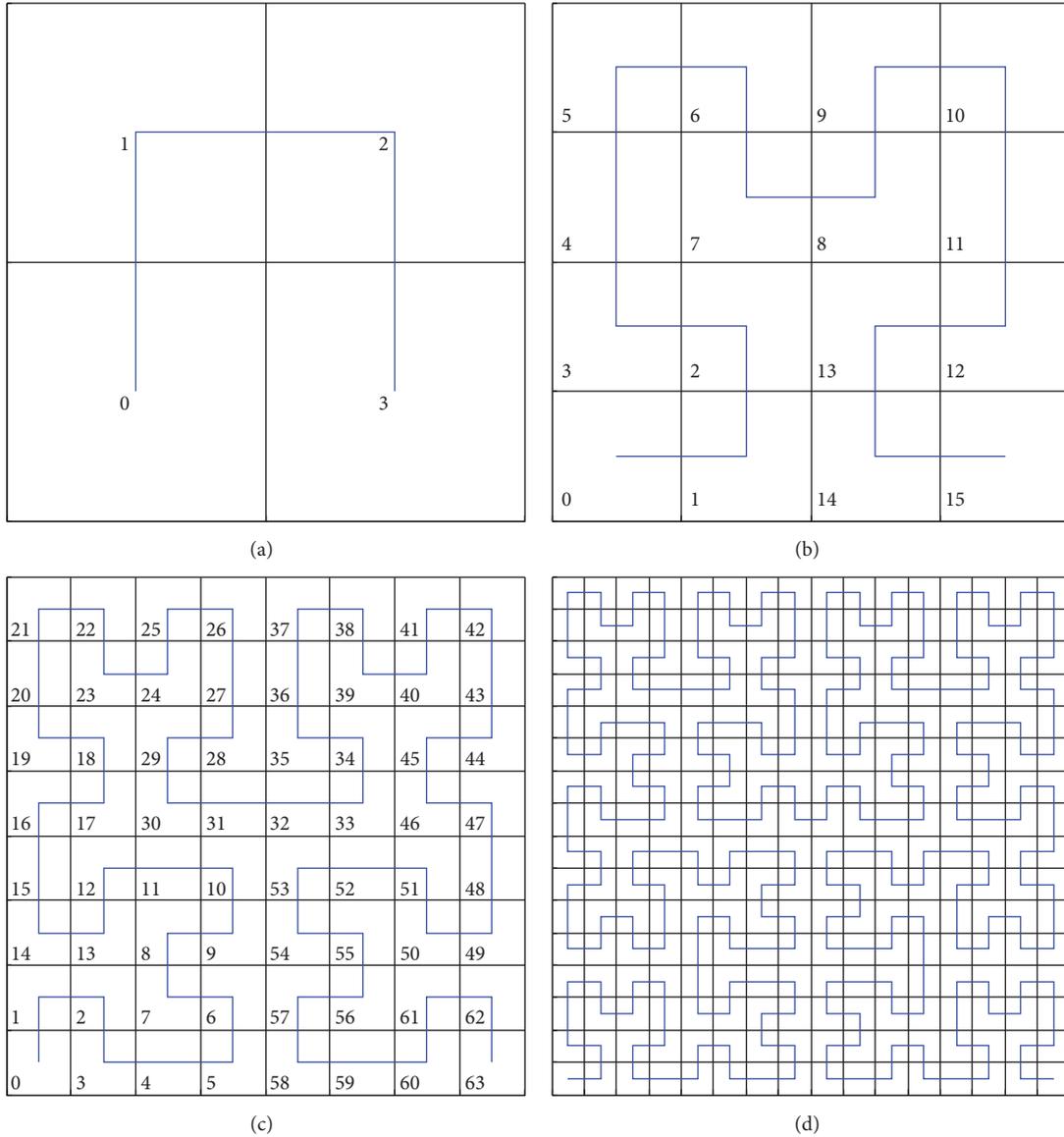


FIGURE 1: (a) Hilbert curve of order $N = 1$. (b) Hilbert curve of order $N = 2$. (c) Hilbert curve of order $N = 3$. (d) Hilbert curve of order $N = 4$.

poor practicality of Gentry’s fully homomorphic encryption scheme, many optimization schemes of the fully homomorphic encryption have emerged since 2009. In 2010, van Dijk et al. [18] applied simple algebraic structure to construct a fully homomorphic encryption scheme, which is based on integer arithmetic, on the basis of Gentry’s scheme. This scheme is simpler, but not practical.

In 2010, Smart and Vercauteren [19] proposed a fully homomorphic encryption scheme with smaller key and ciphertext size. In 2011, Gentry and Halevi [20] improved the original fully homomorphic encryption scheme with a new key generation algorithm, and the full polynomial inversion is not required in this scheme. In 2011, Gentry and Halevi [21] put forward some schemes which did not require squashing step and the hardness of SSSP assumption to further optimize the performance, and the practicability was improved. In

2012, Brakershi et al. [10] proposed the Brakerski-Gentry-Vaikuntanathan homomorphic encryption scheme which applied the key switching and modulus switching technique, meanwhile with additional BitDecomp technology to manage the noise. In the same year, Coron et al. [22] proposed a fully homomorphic encryption scheme over the integers which reduces the public key size of the van Dijk et al. scheme.

Most of the recent research results are still achieved by improving the original Gentry’s scheme. Halevi and Shoup developed a new fully homomorphic encryption library, namely, HELib [23] in 2013, which is based on the BGV homomorphic encryption scheme, using modulus switching technique to reduce the ciphertext noise. It accomplishes the homomorphic operation of subtraction and shift, on the basis of the original additions and multiplications. In addition to

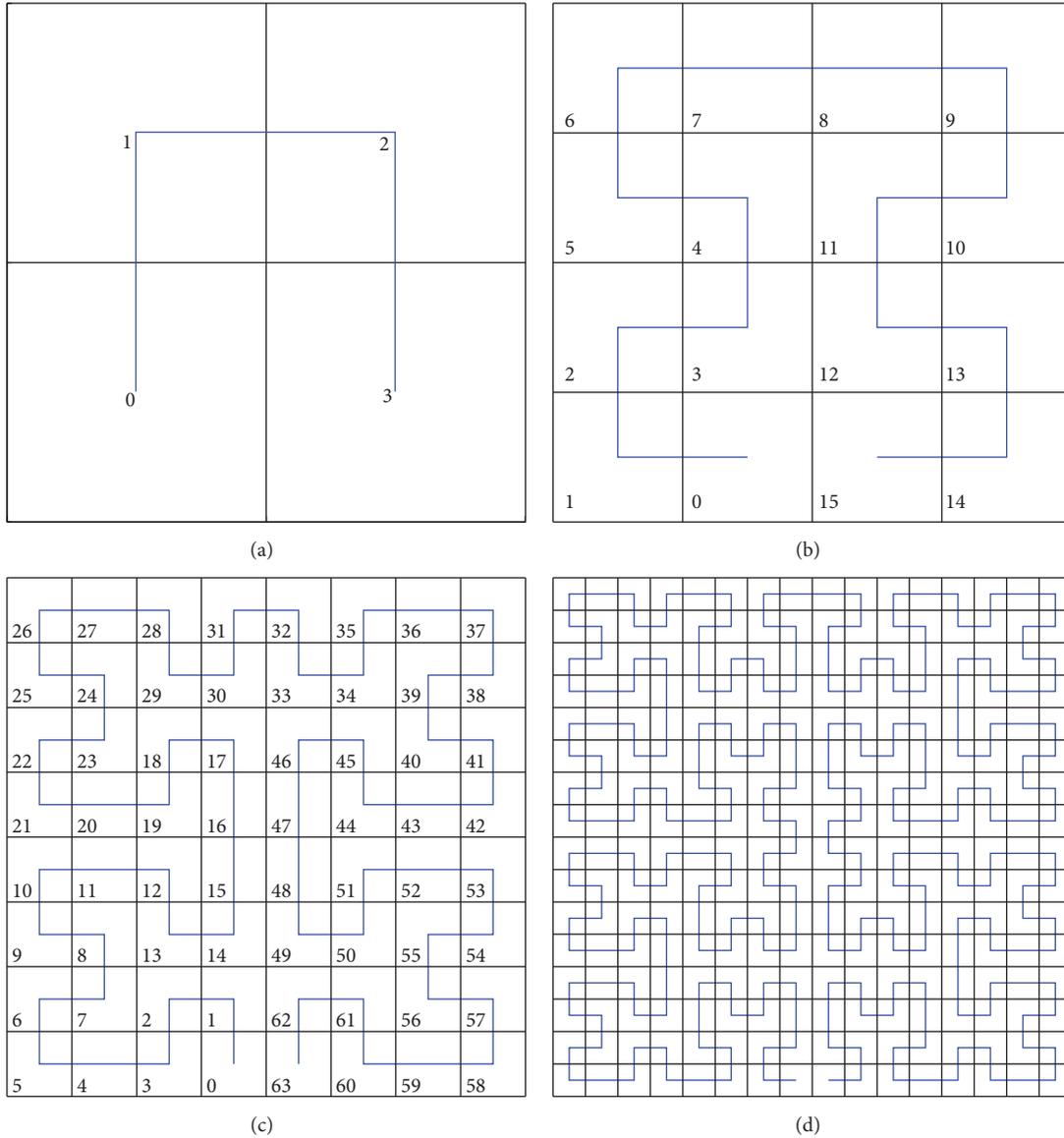


FIGURE 2: (a) Moore curve of order $N = 1$. (b) Moore curve of order $N = 2$. (c) Moore curve of order $N = 3$. (d) Moore curve of order $N = 4$.

the functional improvement, the performance of HELib is mainly optimized by Smart-Vercauteren ciphertext packing technique [24] and Gentry-Halevi optimization [25] to further improve the efficiency of homomorphic operations. In general, the homomorphic operation of encrypted data is accomplished and it shows better performance.

Taking everything above into consideration, we adopt the homomorphic encryption scheme released in HELib in the proposed protocol.

3. The Proposed Protocol

An efficient private circular query protocol is proposed which can achieve high accuracy and low computation cost without the trusted third party.

3.1. Overview of EPCQP. Figure 3 illustrates the whole architecture of EPCQP.

3.1.1. Initialization Process

Step 1. The server constructs a Moore curve and generates the H -index for every registered POI on the target region.

In this step, an LBS server selects the appropriate parameters to construct a Moore curve that covers up the target region and builds the POI-table containing the information of all registered POIs. The POI-table contains the H -index and POI-info of each POI, for example, the longitude, the latitude, and the name. Each stored POI in the POI-table is numbered in accordance with the evenly distributed H -index

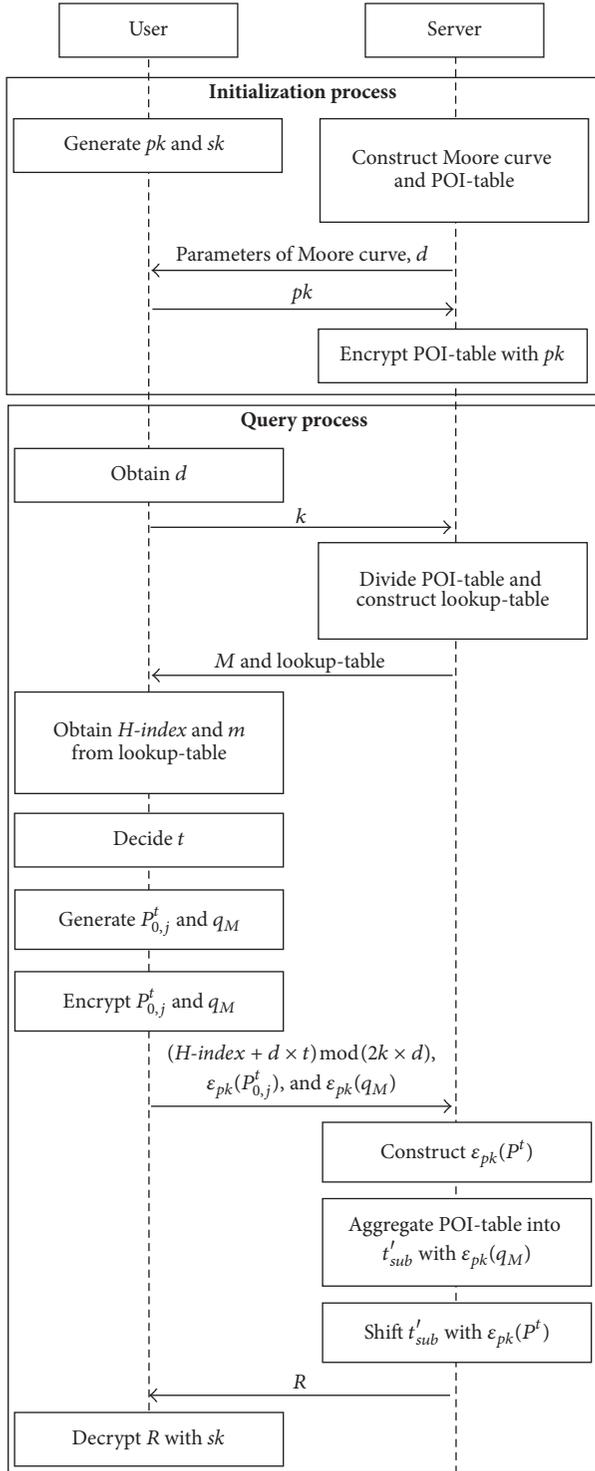


FIGURE 3: Interaction diagram of EPCQP.

with common difference d instead of the associated H -value. The definition of H -index will be presented in Section 3.2.1.

Step 2. The user generates the public and private key pairs and sends the public key pk to the server. Then the server encrypts the POI-table with the public key pk .

3.1.2. Query Process

Step 3. The user issues a k -NN query to the server.

Step 4. The server divides the POI-table into M subtables. Note that the H -index column of the POI-table will not be encrypted. The number of all entries in each subtable is $2k$. $ID_{sub-table}$ denotes an index of the subtable in which a POI is contained. The server stores the mapping from H -value to H -index of each POI into a lookup-table. In addition, the lookup-table records the $ID_{sub-table}$ in order that the user knows which subtable he should search. Consequently, the lookup-table contains three attributes of each POI: H -index, H -value, and $ID_{sub-table}$. The details of the dividing method will be presented in Section 3.2.2.

Step 5. The server announces the setting parameters of the Moore curve and the lookup-table to all registered users in public. The user can retrieve the H -index of his current location on the target region and the subtable that he should search. The details will be presented in Section 3.2.3.

Step 6. The user chooses an integer t to generate a $2k \times 2k$, t -offset circular shift permutation matrix $P_{i,j}^t$ ($i, j = 0, \dots, 2k - 1$) and a vector q_M . In the first row of $P_{i,j}^t$, the $(2k - t + 1)$ -th element is the only nonzero element. The m -th element of q_M is the only nonzero element. The definitions of $P_{i,j}^t$ and q_M will be presented in Sections 3.2.3 and 3.2.4.

Step 7. The user calculates the value $(H\text{-index} + d \times t)$ modulo $(2k \times d)$ to generate the shift- H -index. The H -index is retrieved from the lookup-table according to the current location. The user encrypts q_M and the first row of $P_{i,j}^t$ by the public key pk generated in Step 2, denoted as $\epsilon_{pk}(q_M)$ and $\epsilon_{pk}(P_{0,j}^t)$. Then the user sends the shift- H -index, $\epsilon_{pk}(P_{0,j}^t)$, and $\epsilon_{pk}(q_M)$ to the server.

Step 8. The server utilizes $\epsilon_{pk}(P_{0,j}^t)$ to construct a secret circular shift matrix $\epsilon_{pk}(P^t)$. The server aggregates all of the subtables with $\epsilon_{pk}(q_M)$ into a new table t'_{sub} . The H -indexes of the POIs in t'_{sub} are numbered again from d to $(2k \times d)$.

Step 9. The server utilizes $\epsilon_{pk}(P^t)$ to perform a secret circular shift on t'_{sub} based on the fully homomorphic property with the public key of the user. The server performs a k -NN search upon the circularly shifted t'_{sub} and then returns the k encrypted results to the user. The detail will be present in Section 3.2.4.

Step 10. The user decrypts the received results with the private key sk selected in Step 2 and obtains the required k -NN POIs.

The H -index of the user has been added by t (in Step 7), and the POIs in t'_{sub} have been secretly circularly shifted by $(d \times t)$ (in Step 9). Based on the additive and multiplicative homomorphism, the secret k -NN search results in the shifted

t'_{sub} will be consistent with the results searched in their plaintexts of the original subtable.

3.2. Details of EPCQP

3.2.1. Mapping from H-Value to H-Index. Figure 1 shows that the starting cell is not adjacent to the ending cell in the Hilbert curve. The searching range will be reduced when the user is near to the starting or ending cell of Hilbert curve, and it means that the query accuracy will be reduced. Figure 2 indicates that the start point and the end point of Moore curve are neighbors. Therefore, we adopt Moore curve to transform a two-dimensional space into a sequence of *H-values*. With the capability of Moore curve, all the POIs can be constructed into a circular structure which is important when dividing the POI-table in the subsequent steps. In addition, the results mainly rely on the order of *H-values*, so that altering *H-values* of POIs do not affect the query results.

H-index denotes an evenly distributed sequence numbered in the ascending order of *H-values* with a common difference d . *H-index* of the i -th POI in a Moore curve is calculated as

$$H\text{-index}(i) = d \times i, \quad (2)$$

where d is an integer greater than or equal to one and i is the sequencing-order of the POI along with the ascending order of *H-values* in the given Moore curve. The server constructs a POI-table which contains the POI-info and *H-index* of all POIs and records the mapping from *H-value* to *H-index* in the lookup-table. The server should update the POI-table and lookup-table whenever any POI changes and then publicly announces the new lookup-table to all registered users.

3.2.2. Dividing the POI-Table. PCQP requires a huge number of calculations due to the multiplication applied across the entire POI-table. Utsunomiya et al. proposed a lightweight k -NN search protocol according to dividing the POI-table into M subtables. The server divides a POI-table in the initialization process in advance in LPCQP. When the user issues a k -NN query, he selects only one subtable which contains POIs that the user needs. Although LPCQP reduces computation cost observably compared with PCQP, it requires a trade-off between the query accuracy and computation cost on the server. Utsunomiya et al. have illustrated that k/n_p should be 0.5 or less in order to obtain the high query accuracy, where n_p denotes the number of all entries in a subtable. The query accuracy of LPCQP becomes worse as M get larger due to the loss of some POIs from aggregating subtables. Extremely, when $k > n_p$, the server only returns n_p POIs to the user. The dividing of the POI-table is performed only once in advance; therefore, it cannot satisfy the query requirement if M is not appropriate.

Inspired by LPCQP, we propose an efficient k -NN search scheme to mitigate the drawbacks of LPCQP, called EPCQP. In EPCQP, the size of each subtable depends on the query requirement of the user instead of dividing the POI-table

only once in the initialization process. We divide the POI-table into M subtables, and each subtable has $2k$ POIs. M is calculated as

$$M = \left\lceil \frac{n}{2k} \right\rceil, \quad (3)$$

where n denotes the number of all entries in the POI-table. The j -th subtable, denoted as t'_{sub}^j , is defined as

$$t'_{\text{sub}}^j = [I_i^j]_{1 \leq i \leq 2k} \quad (4)$$

$$I_i^j = I_{2k \times (j-1) + i},$$

where I_i denotes the i -th entry of the original POI-table and I_i^j denotes the i -th entry of the j -th subtable.

As mentioned in Section 3.2.1, the start point is adjacent to the end point in Moore curve. Geographically, the first POI and the last POI stored in the POI-table are close to each other. Therefore, the first and the last POIs are neighbors in two-dimensional space regardless of the *H-index* distance between them. If the number of all entries in the M -th subtable is less than $2k$, it will be appended using the entries in the original POI-table from the first to the l_p -th in order. l_p is calculated as

$$l_p = 2k \times \left\lceil \frac{n}{2k} \right\rceil - n. \quad (5)$$

As shown in Figure 4, there are nine POIs in the POI-table. If $k = 2$, the POI-table will be divided into three subtables each containing four POIs. The third subtable contains i , a , b , and c .

3.2.3. Aggregating Subtables. The user looks for the *H-index* from the lookup-table according to the current location and then chooses the subtable which contains the nearest POI as the subtable for querying. Next, the user obtains the index of the target subtable by retrieving the $\text{ID}_{\text{sub-table}}$ column of the lookup-table. Without loss of generality, let the m -th subtable be the one that the user selects. The user generates a vector q_M defined as

$$q_M = |q_i|_{1 \leq i \leq M}$$

$$q_i = \begin{cases} 1, & i = m \\ 0, & \text{otherwise.} \end{cases} \quad (6)$$

The user encrypts q_M with the public key pk and then sends the ciphertext $\varepsilon_{\text{pk}}(q_M)$ to the server. The server multiplies each element of $\varepsilon_{\text{pk}}(q_M)$ by the corresponding subtable. Then the j -th subtable t'_{sub}^j becomes t'^j_{sub} .

$$t'^j_{\text{sub}} = \varepsilon_{\text{pk}}(q_j) \times_c \varepsilon_{\text{pk}}(t'^j_{\text{sub}})$$

$$= [\varepsilon_{\text{pk}}(q_j) \times_c \varepsilon_{\text{pk}}(I_i^j)]_{1 \leq i \leq 2k}. \quad (7)$$

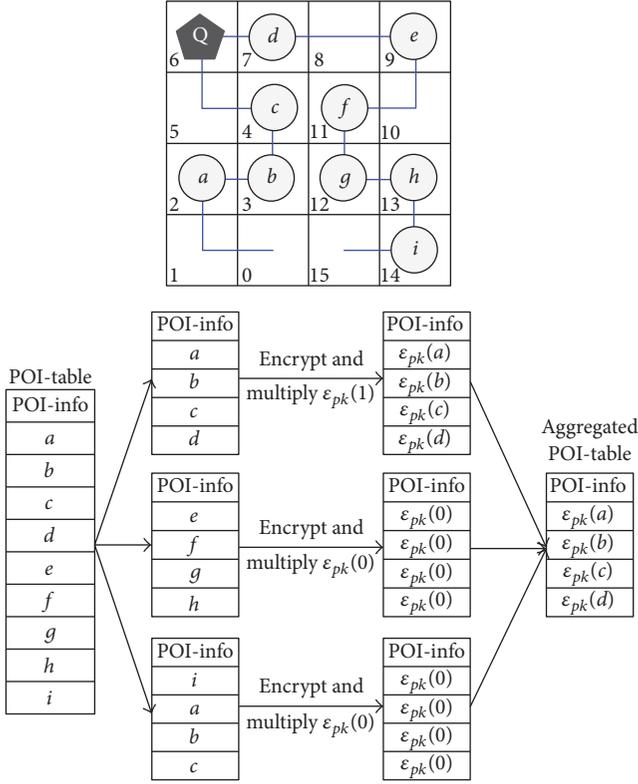


FIGURE 4: Example of aggregating subtables.

The server aggregates all the subtables into a new table t'_{sub} calculated as

$$t'_{\text{sub}} = t'_{\text{sub}}{}^1 +_c t'_{\text{sub}}{}^2 +_c \dots +_c t'_{\text{sub}}{}^M = [I'_i]_{1 \leq i \leq 2k}$$

$$I'_i = \varepsilon_{\text{pk}}(q_1) \times_c \varepsilon_{\text{pk}}(I_i^1) +_c \varepsilon_{\text{pk}}(q_2) \times_c \varepsilon_{\text{pk}}(I_i^2) \quad (8)$$

$$+_c \dots +_c \varepsilon_{\text{pk}}(q_M) \times_c \varepsilon_{\text{pk}}(I_i^M).$$

According to (6) and (8), all entries of the POI-info in the subtables become zero in their plaintext domain except the m -th subtable. Note that the H -indexes of the POIs in t'_{sub} are numbered from d to $(2k \times d)$.

Due to the properties of homomorphic encryption, t'_{sub} satisfies

$$D_{\text{sk}}(I'_i) = I_i^m. \quad (9)$$

Figure 4 indicates the process of aggregating subtables. There are nine POIs and a user (Q) on the map. According to the Moore curve, the nine POIs are stored in the POI-table in the ascending order of H -indexes. When the user issues a k -NN search with $k = 2$, the server divides the POI-table into three subtables and each subtable has four POIs. The POI d has the nearest H -index to the user; therefore, the user selects the first subtable to query. Then all the subtables are aggregated into one table with the entries of the first subtable.

3.2.4. Secret Circular Shift in the m -th Subtable. In order to keep the original H -index secret to the server, the POI-info column of the original POI-table is circularly shifted in PCQP. On the basis of Paillier encryption scheme, an approach for circular shift by the encrypted matrix-vector multiplication is proposed in PCQP, where the POI-info column is in its plaintext domain whereas the permutation matrix is encrypted. Although the circular shift in the entire POI-table maintains the neighboring relationship with POIs, it requires a huge number of calculations due to the multiplication applied across the entire POI-table. Utsunomiya et al. proposed a lightweight scheme (LPCQP) to mitigate the drawbacks of PCQP. Inspired by LPCQP, we utilize the circular shift in t'_{sub} and modulo operation to hide the real location of the user.

After shifting the POI-info column t units downward circularly, the same k -NN search results can be obtained by changing the querying H -index to the shift- H -index, which is calculated as

$$\text{shifted-}H\text{-index} = (H\text{-index} + d \times t) \bmod (2k \times d). \quad (10)$$

When t is a negative integer, it represents an upward shifting.

The shift parameter t is decided by the user; hence the server cannot obtain the original H -index of the user according to the shifted- H -index. The current location of the user is protected from being disclosed.

Likewise, the circularly shift on the POI-info column on the server side should be kept secretly. In PCQP, the entire POI-info column has been circularly shifted by multiplying an encrypted $n \times n$ matrix. The operator of multiplicative and additive homomorphism will incur a huge overhead, especially for the mobile services. In this paper, only the POI-info of the m -th subtable which the user selects will be circularly shifted by multiplying an encrypted $2k \times 2k$, t -offset matrix $P^t_{i,j}$, which is defined as

$$P^t_{i,j} = \begin{cases} 1, & j = i + (2k - t) \bmod (2k) \\ 0, & \text{otherwise,} \end{cases} \quad (11)$$

where $i, j = 0, 1, \dots, 2k - 1$.

The user encrypts the permutation matrix $P^t_{i,j}$ with the public key pk to obtain the ciphertext $\varepsilon_{\text{pk}}(P^t)$ and then sends it to the server. The server multiplies $\varepsilon_{\text{pk}}(P^t)$ with the aggregated table t'_{sub} in order to circularly shift the POI-info column of the m -th subtable and keeps the H -index column intact.

Note that the pseudorandom numbers are different during each encryption process, it means that the server has no way to distinguish the encrypted '0' and '1'. In order to reduce the computation and communication cost, the user only encrypts the first row of $P^t_{i,j}$ and sends $\varepsilon_{\text{pk}}(P^t_{0,j})$ to the server. Then sever constructs $\varepsilon_{\text{pk}}(P^t)$ by circularly shift on $\varepsilon_{\text{pk}}(P^t_{0,j})$. For ease of understanding, there is an example on how to construct $\varepsilon_{\text{pk}}(P^t)$.

H-index	POI-info
2	a
4	b
6	c
8	d
10	e
12	f
14	g
16	h

(a)

H-index	POI-info
2	c
4	d
6	e
8	f
10	g
12	h
14	a
16	b

(b)

FIGURE 5: (a) The evenly distributed POIs. (b) After circularly shifting, the shift- H -index = $(10 + 2 \times 6) \bmod 16 = 6$.

Let $(\varepsilon_{\text{pk}}(P_{00}^t) \ \varepsilon_{\text{pk}}(P_{01}^t) \ \varepsilon_{\text{pk}}(P_{02}^t) \ \varepsilon_{\text{pk}}(P_{03}^t))$ be the first row of $\varepsilon_{\text{pk}}(P^t)$, and $\varepsilon_{\text{pk}}(P^t)$ can be constructed as

$$\begin{pmatrix} \varepsilon_{\text{pk}}(P_{00}^t) & \varepsilon_{\text{pk}}(P_{01}^t) & \varepsilon_{\text{pk}}(P_{02}^t) & \varepsilon_{\text{pk}}(P_{03}^t) \\ \varepsilon_{\text{pk}}(P_{03}^t) & \varepsilon_{\text{pk}}(P_{00}^t) & \varepsilon_{\text{pk}}(P_{01}^t) & \varepsilon_{\text{pk}}(P_{02}^t) \\ \varepsilon_{\text{pk}}(P_{02}^t) & \varepsilon_{\text{pk}}(P_{03}^t) & \varepsilon_{\text{pk}}(P_{00}^t) & \varepsilon_{\text{pk}}(P_{01}^t) \\ \varepsilon_{\text{pk}}(P_{01}^t) & \varepsilon_{\text{pk}}(P_{02}^t) & \varepsilon_{\text{pk}}(P_{03}^t) & \varepsilon_{\text{pk}}(P_{00}^t) \end{pmatrix}. \quad (12)$$

The server multiplies $\varepsilon_{\text{pk}}(P^t)$ with the aggregated table t'_{sub} to obtain the circularly shifted and decrypted POI-info data. After decryption, the results will have the same shifted value t .

As shown in Figure 5(a), there are 8 POIs in the subtable, where $k = 4$ and $d = 2$. The H -index of the user is 10, and the results will be c , d , e , and f . The shift- H -index is 6 when $t = 6$. As presented in Figure 5(b), the user will obtain the same results according to the shift- H -index.

Note that the first entry and the last entry of the subtable are not adjacent to each other. Therefore, the neighboring relationship will be changed after circularly shifting the m -th subtable, and this will reduce the search accuracy. This effect is negligible compared with the reduction in computation and communication overhead.

3.2.5. k -NN Search. Lien et al. [8] proposed a cross-like k -NN search algorithm to achieve high accuracy rate and showed that two additional queries started from the central cells of search region are sufficient to achieve the reasonable accuracy rate in most cases. However, it may include duplicated POI especially when the user is close to the boundary. Utsunomiya et al. [9] proposed a group-based query point selection algorithm which achieved a higher accuracy rate. We adopt the methods proposed in LPCQP to improve the accuracy rate.

3.3. Security Analysis. Mainly the most common attacks that can obtain some private information of the user can be listed as the following 6 chances. We divide them into three groups where there are two of them in each group.

The first group consists of correlation attack and background knowledge attack. The adversary utilizes the former

one to eavesdrop some input queries and output results through the network. Then combining with some prior knowledge obtained from the latter one about the basic information of the user such as age or job, the adversary can infer the location of the user with a relatively large probability.

The second group includes offline keyword guessing attack as well as inference attack. There is a trapdoor generated from a search word and it does not leak any information of this search word. The first attack guesses the content of the encrypted data by computing trapdoors of some widely used words. Simultaneously, the second attack combines some background knowledge of the data content with some access patterns to identify the trapdoor of some words.

The last group is divided into man-in-the-middle attack (MITM) and link attack. For MITM, the adversary needs to be a third party between users and servers. It needs to guarantee that both two parties believe they are having conversations directly with the other one. When it comes to the link attack which is more commonly used, the adversary combines the inaccurate location information of the user and the data source from the outside to determine the accurate location or the identification of the user.

In this paper, we are aiming at proposing an efficient scheme for k -NN search with perfect privacy-preserving. In the following part, we will illustrate the security analysis of six different kinds of attacks that are mentioned previously.

3.3.1. The Correlation Attack. Correlation attack belongs to the plaintext attacks that utilize a statistical weakness due to a poor choice of the Boolean function.

The correlation attacks can be successfully mounted due to the fact that obvious correlations between the output of a special linear feedback shift register (LFSR) and the outputs of all the LFSRs defined by Boolean functions can be distinguished. Thus combining with part of the keystream knowledge, an adversary can get the key of the special LFSR by brute-forcing.

As for the correlation attack in EPCQP, the user sends $\varepsilon_{\text{pk}}(P_{0,j}^t)$ and $\varepsilon_{\text{pk}}(q_M)$ which are all encrypted with distinct random numbers, with these random numbers being changed every time. Therefore, the server will have no idea of correlating queries issued by the user. It implies that our scheme is secure under the correlation attack.

3.3.2. The Background Knowledge Attack. This kind of attack exploits the close relationship between several standard common attributes where the sensitive attribute is exactly among them. In this way, the adversary can reduce the cardinality of the possible value set to find this sensitive attribute.

It can be seen that only the user knows the shifts amount because it is encrypted in our scheme. As for the server, it can learn nothing about the location though it can get the query history and the profiles of the user. This arises from the fact that the user can change the shifts amount every time while all the sensitive information transferred is encrypted.

In terms of this attack in our scheme, we only transfer the shifted location chosen by the user during the k -NN search. The shifted location $(H\text{-index} + d \times t) \bmod (2k \times d)$

is not sensitive because the shifts amount is selected by the user independently and randomly. Therefore, no sensitive information about the user will be leaked to the server, and thus our scheme is resistant to this attack.

3.3.3. The Offline Keyword Guessing Attack. Generally speaking, dictionary attack as well as offline guessing attack occurs based on the fact that the so-called weak secrets may have low entropy. This means that it comes from the value set of a small cardinality. Similarly, the keywords also come from a rather smaller set when compared with the weak secrets such as passwords. Basically, low entropy means high probability, so the user is more likely to use the low entropy keywords when querying the table.

In our scheme, as for the offline keyword guessing attack, taking the location of the user, for example, the user rarely uses the low entropy keywords. Particularly, the location data in the server is collected as a lookup-table which is not the sensitive information of the user. Accordingly, by offline keyword guessing attack, no one can guess location data of the user during the query in our scheme.

3.3.4. The Inference Attack. The inference attack aims at gaining knowledge of the user by analyzing the data. If the adversary can obtain the real value of the sensitive information with a relatively high probability, we say the user leaks the information. In the whole process, the adversary cannot directly obtain any data from some trivial information.

When it comes to the inference attack in our scheme, it utilizes the access pattern of the user, such as a document that contains previously queried keywords. However, no sensitive information about the access pattern of the user will be leaked to the server or an adversary because $P_{i,j}^t$ and q_M are encrypted with distinct random numbers, and these random numbers are scrambled by the user each time when the user queries. Consequently, our scheme is robust to this attack.

3.3.5. The Man-in-the-Middle Attack. The man-in-the-middle attack is a widely used attack where there exists a third party called man-in-the-middle. It plays the role of a user when communicating with the server and then it will play the role of a server when facing the user. So it needs to imitate all the information transmitted in the network to make the user and the server believe that they are truly having conversation with the other one.

There are two widely used methods in preventing man-in-the-middle attack. Authentication is used to make sure that the information transmitted during the communication is from a legitimate source. Tamper detection ensures that the information is not tampered during the transmission.

In our scheme, the user will never send any sensitive information or plaintext to the server, nor will the server do. The user only sends k , pk , $(H\text{-index} + d \times t) \bmod (2k \times d)$, $\varepsilon_{pk}(P_{0,j}^t)$, and $\varepsilon_{pk}(q_M)$. As for the server, it only needs to send the parameters of Moore curve, lookup-table, d , M , and the final result set which is encrypted by the homomorphic encryption scheme. Totally, they are encrypted with distinct random numbers and the numbers are varied every time

when the user queries. Meanwhile the information of m which is the location of the subtables is transmitted in the form of $\varepsilon_{pk}(q_M)$. Thus, the adversary cannot play the role of the server, because he has no idea of m and cannot communicate with the user. In sum, our scheme can defend against this attack.

3.3.6. The Link Attack. When it comes to the link attack which is more commonly used, the adversary combines the inaccurate location information of the user and the data source from the outside to determine the accurate location or identification of the user. However, all of the location information is represented as $(H\text{-index} + d \times t) \bmod (2k \times d)$ in the aggregated subtable. The server can learn nothing from the shift amount, though it may have access to the partial knowledge of the data source from the outside. Therefore, our scheme can also guarantee the security under link attack.

4. Performance Evaluation

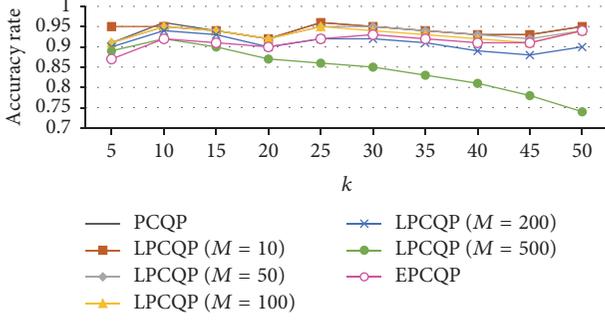
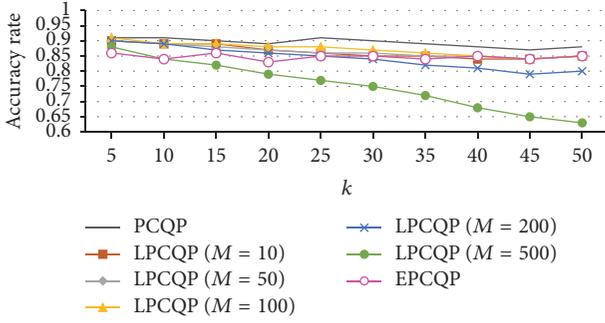
In this section, we compare the performance of EPCQP with that of the related two works: PCQP [8] and LPCQP [9]. We adopt the homomorphic encryption scheme released in HELib. The proposed scheme is implemented in JAVA language and performed on a laptop computer with a 1.6 GHz Intel Core i5 CPU and 4 GB RAM.

We use two datasets including a uniform dataset and a real-world dataset. Each dataset contains 10000 POIs. The real-world dataset is extracted from the base stations datasets in China. We randomly select 1000 locations on the map to issue the k -NN search in each experiment and the results are averaged.

4.1. Query Accuracy. The accuracy criteria have been well used in data mining and machine learning areas [26–30]. Similarly, query accuracy is used for validation of the experiment in this paper. The value of k and M is varied from 5 to 50 and from 10 to 500, respectively. Let R and G denote the returned result set and the k -NN ground-truth result set, and the accuracy rate is defined as

$$r_{\text{acc}} = \frac{|R \cap G|}{|G|}. \quad (13)$$

In LPCQP, only the m -th subtable is selected by the user for searching. The number of the subtables will directly affect the query accuracy. When k is greater than n_p , the server returns less than k POIs to the user. Although the computation cost is reduced by dividing the POI-table only once in the initialization process, the query accuracy rate decreases in some cases. After dividing the POI-table into M subtables, $(n - n_p)$ POIs are lost due to the aggregating subtables in (8). Hence, the quality of the results gets even worse as the number of the subtables gets larger. In contrast, the server can construct the subtables with a large unit when M is small, which guarantees that the number of results satisfies the querying requirement. Nevertheless, when n_p is far greater than k , as mentioned in (7) and (8), the computation cost incurred by the homomorphic additions

FIGURE 6: Query accuracy rate versus k of the uniform dataset.FIGURE 7: Query accuracy rate versus k of the real-world dataset.

and multiplications will be too wasteful to find at most k POIs. In order to find the maximum M that satisfies the high accuracy and significantly reduces the computation cost, Utsunomiya et al. [9] defined the ratio α as

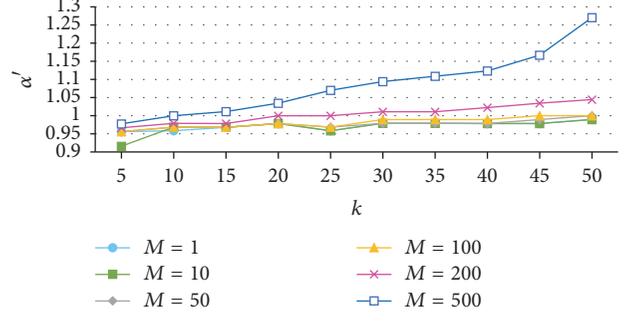
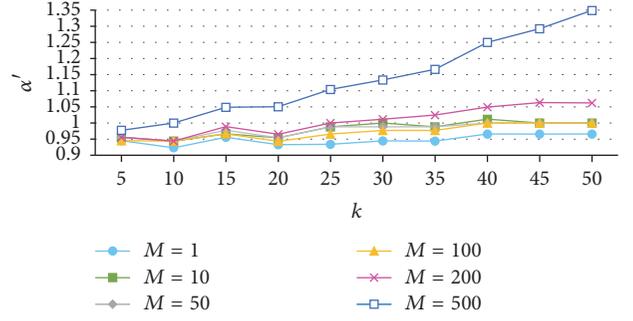
$$\alpha = \frac{r_{\text{LPCQP}}}{r_{\text{PCQP}}}, \quad (14)$$

where r_{LPCQP} and r_{PCQP} denote the accuracy rate of LPCQP and PCQP. The performance results showed that k/n_p should be 0.5 or less in order to achieve $\alpha \geq 0.95$. It means that when $n_p \geq 2k$, the scheme can keep a considerable high accuracy and reduce the computation cost.

When $k > n_p/2$, the entries lost from aggregating the subtables are too much. The situation will get worse when $k > n_p$, and the user will receive less than k results in this case.

In our scheme, $n_p = 2k$. As mentioned above, EPCQP achieves a high query accuracy rate and reduces the computation cost which is presented in Section 4.2. The advantage of EPCQP is obvious compared with the case that $k > n_p$ in LPCQP.

Figures 6 and 7 represent the accuracy rate versus k for the different datasets. It shows that the accuracy rate of EPCQP is higher than that of LPCQP when M is 500. Although the accuracy rate of EPCQP is slightly lower than that of PCQP and LPCQP when $M \leq 100$, it is still higher than that of LPCQP when $k > 25$ and $M = 200$. The accuracy rate of EPCQP is higher than 90% even if k is large for the uniform

FIGURE 8: α' versus k of the uniform dataset.FIGURE 9: α' versus k of the real-world dataset.

dataset, and it is higher than 84% when k is large for the real-world dataset. We define α' as

$$\alpha' = \frac{r_{\text{EPCQP}}}{r_{\text{LPCQP}}}, \quad (15)$$

where r_{EPCQP} denotes the accuracy rate of EPCQP. Figures 8 and 9 indicate the ratio α' versus k for the two datasets. When $M = 1$, the value of α' denotes the ratio of r_{EPCQP} to r_{PCQP} . Note that the ratio α' is kept to 0.9 or above regardless of the various k for the two datasets. Particularly, for the cases of $M = 500$, EPCQP achieves $\alpha' \geq 1$ when $k \geq 10$. As shown in Figures 8 and 9, EPCQP keeps high accuracy rate of LPCQP when $M < 500$ and improves the accuracy rate when $M = 500$. Besides, the advantage of EPCQP will be more significant when M is larger than 500.

4.2. Computation Cost. In the query process of EPCQP, the server first divides the POI-table into M subtables. Compared with calculation of ciphertexts and homomorphic encryption/decryption, the cost of dividing table is negligible. As presented in Section 3.2.3, the server multiplies each element of $\varepsilon_{pk}(q_M)$ by the POI-info column of the corresponding subtable and then aggregates all the subtables into a table t'_{sub} . The process of aggregating requires n multiplication and $2k(M - 1)$ additions according to (8). Finally, the server multiplies $\varepsilon_{pk}(P')$ with the aggregated table t'_{sub} to obtain the circularly shifted and decrypted POI-info data which requires $k(2k - 1)$ additions and $(k \times 2k)$ multiplications.

Table 1 represents the computation cost of PCQP, LPCQP, and EPCQP. Compared with the cost of shifting process, the cost on aggregating subtables can be negligible when k is

TABLE 1: Comparison of computation cost on the server.

Scheme	Addition	Multiplication
PCQP	$k(n-1)$	kn
LPCQP	$n_p(M-1) + k(n_p-1)$	$n + kn_p$
EPCQP	$2k(M-1) + k(2k-1)$	$n + 2k \times k$

n and n_p denote the number of all entries in the POI-table and a subtable, respectively.

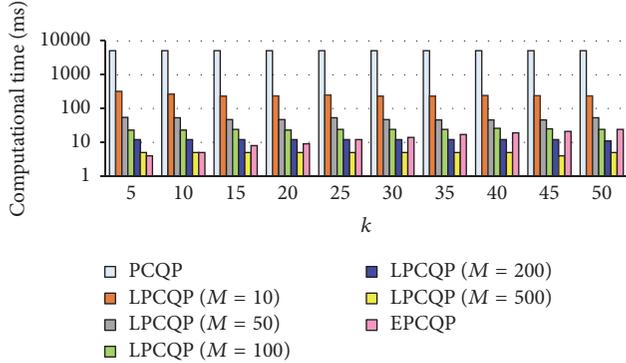


FIGURE 10: Computational time for encrypting the circular shift matrix.

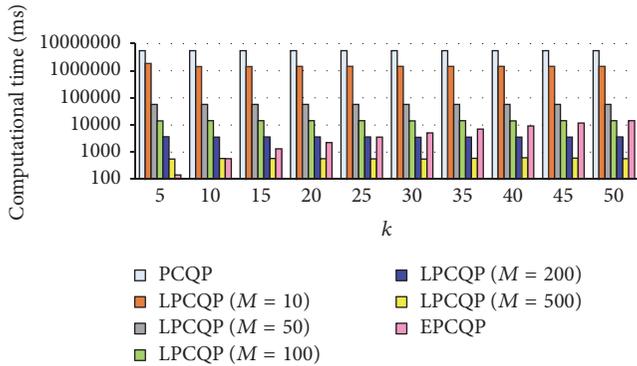


FIGURE 11: Computational time for the shifting process.

large. As mentioned in Section 4.1, in order to achieve a high accuracy in LPCQP, n_p should be greater than or equal to $2k$. As shown in Table 1, the cost of multiplication in EPCQP is lower than that in LPCQP when $n_p > 2k$. Besides, the cost of addition in the shifting process is reduced obviously. Therefore, our proposed scheme has a high accuracy with the lower computation cost.

The proposed scheme has the approximate computational time for aggregating subtables compared with that in LPCQP. Figures 10 and 11 represent the computational time for encrypting the circular shift matrix and the shifting process, respectively. As shown in Figure 10, the computation cost on encrypting the circular shift matrix in EPCQP becomes one-thousandth of that in PCQP or below when $k \leq 20$, and it is about one-tenth of that in LPCQP regardless of the various k when $M = 10$. Figure 11 shows that the computation cost of the shifting process in EPCQP becomes one-thousandth of

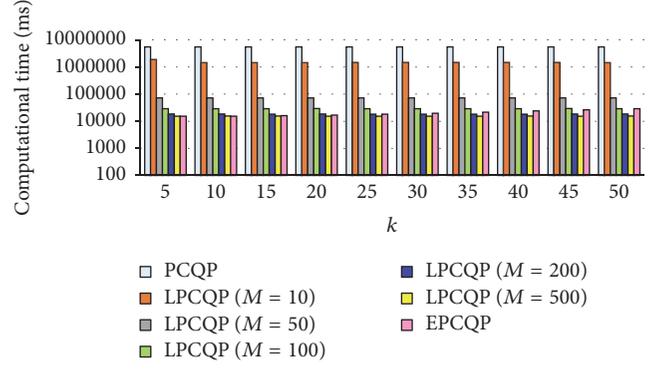


FIGURE 12: Total computation cost.

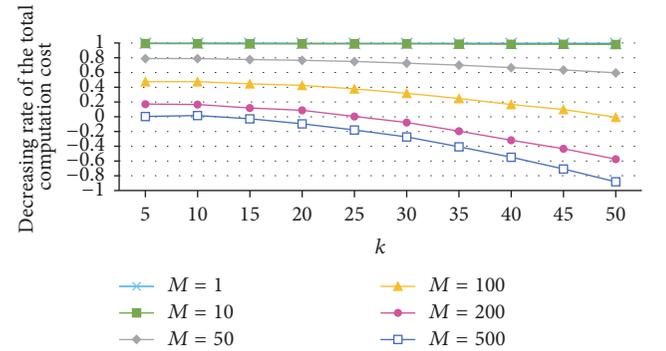


FIGURE 13: Decreasing rate of the total computation cost.

that in PCQP or below when $k \leq 30$, and it is one percent of that in LPCQP regardless of the various k when $M = 10$. Figures 10 and 11 show that the computation cost of EPCQP is lower than that in LPCQP when $M \leq 100$. Although the computation cost of EPCQP is slightly higher than that in LPCQP when $M = 200$ and $k > 25$, the accuracy rate of EPCQP is higher than that of LPCQP. Figure 12 represents the total computation cost of the PCQP, LPCQP, and EPCQP. As shown in Figure 12, the computation cost of our scheme is lower than that of LPCQP regardless of k when $M \leq 100$. The decreasing rate of the total computation cost is represented in Figure 13. When $M = 1$, it denotes the decreasing rate of the total computation cost compared with that of PCQP. From these results, the computation cost of our proposed scheme is reduced by 99% or more compared with that of PCQP. When $k \leq 25$ and $M \leq 200$, the decreasing rates are positive number, and it means that the computation cost of EPCQP is lower than that of LPCQP.

Based on the above result, we can say that EPCQP keeps high accuracy rate of LPCQP while reducing the computation cost. There is a trade-off between the accuracy rate and the computation cost of EPCQP.

4.3. Communication Cost. In this section, we discuss the communication cost of our scheme. Table 2 shows the communication cost of PCQP, LPCQP, and EPCQP. As described in Section 3.2.4, the encrypted matrix is constructed from the first row. Let l denote the bit-length of an encrypted POI-info.

TABLE 2: Comparison of communication cost.

Scheme	Uplink	Downlink
PCQP	$l \times n$	$l \times k$
LPCQP	$l(n_p + M)$	$l \times k$
EPCQP	$l(2k + M)$	$l \times k$

n and n_p denote the number of all entries in the POI-table and a subtable, respectively.

The user sends $\varepsilon_{pk}(P_{0,j}^t)$ as $(l \times n)$ bits to the server in PCQP. In LPCQP, the user sends $\varepsilon_{pk}(P_{0,j}^t)$ and $\varepsilon_{pk}(q_M)$ as $l(n_p + M)$ bits. The communication cost of our scheme is $l(2k + M)$ bits. After searching, the server returns k results to the user.

As showed in Table 2, the downlink communication cost of EPCQP is the same as PCQP and LPCQP. In the uplink process, the communication cost of EPCQP is lower than that of PCQP. In summary, the communication cost of our scheme is in the same degree compared to LPCQP; meanwhile it achieves high accuracy.

5. Related Work

The existing methods for location privacy protection mainly fall into three categories.

5.1. Spatial Cloaking. Spatial cloaking [31–39] methods generate a cloaking region to the location server and the server returns the query results to users or a trusted third party. K -anonymity is the most common model [40], which was firstly implemented in LBS by Gruteser and Grunwald [1]. Chow et al. [5] maintains the location information of the user by using the R-tree structure, which is the classic method for protecting the location of users by using the k -anonymity model. It proposes a scheme that accurately searches k -nearest neighbors in the rectangular anonymous region.

5.2. Location Obstruction. The user continues to submit queries with a specific fake location to the location-based server, and the server iteratively returns the results based on the fake location until the user obtains the result that satisfies the privacy and security requirements. The classic algorithm of location obstruction is SpaceTwist [41], which requires multiple rounds of communication, and the communication cost required for each complete query is large.

5.3. Spatial Transformation. The principle of spatial transformation for protecting location privacy is to convert the location information from conventional data space to another. The scheme [42] utilizes homomorphic encryption to accomplish data interaction between users and servers. Although it achieves strong privacy protection, it is difficult to be adapted to the application environment of continuous queries and real-time responses with extremely expensive computation cost. In order to reduce the computation cost, a scheme [6] based on Hilbert curve is proposed. The scheme, which effectively reduces the computation cost of encryption, transforms all POIs in two-dimensional space into a sequence

of integers in one-dimensional sequence and maintains the original neighborhood relationship approximately. The drawback of this scheme is that the query accuracy is not high. A k -anonymous spatial region construction mechanism [7] is proposed for distributed systems. It combines the user location with Hilbert-order to form a spatial area with other peer nodes and then sends the area and the query requirement to the server. HilAnchor scheme [43] is based on SpaceTwist and Hilbert curve. With only two rounds of communication, the user can get the exact k -nearest neighbor POIs without the leakage of the location. The MobiCrowd algorithm [44] utilizes the buffer to guarantee that queries can be accomplished locally in order to reduce the communication cost.

Query privacy is as important as location privacy. The scheme [45] generates dummy queries so that the server and the attacker cannot obtain the preference information of the user by summarizing the rules of the query contexts. An attack mode [46] marks a query according to the query context, the location, and the querying time. Reference [47] proposed the k -Approximate Beyond Suspicion scheme, which first utilizes a clustering algorithm (such as the K -means) to cluster the users who have the similar location and issue the similar queries and then calculates the anonymous areas according to k -anonymous to protect the privacy of the location and query.

6. Conclusion

In this paper, we propose a privacy-preserving circular query protocol with high accuracy and low complexity, which can be utilized in the location-based k -NN search. With the circular shift and the homomorphic encryption, the proposed scheme accomplishes the efficient querying and the privacy protection simultaneously. We adopt the method that the server dynamically divides the encrypted POI-table according to the query of the user. Our scheme mitigates the drawbacks of PCQP and LPCQP without impairing the advantages of them. The computation cost is reduced by 99% or more compared with that of PCQP by allowing a 6% reduction in the accuracy rate regardless of k . Comparing with LPCQP when $M \leq 100$, the computation cost is reduced by up to 47.5% with a 5% reduction in the accuracy rate. With the rapid development of the spatial crowdsourcing, the location privacy has attracted more and more attention. We expect our scheme will inspire the research of location privacy protection and encrypted data computing in spatial crowdsourcing.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was supported by the National Key Research and Development Program of China under Grants

2017YFB0802202 and 2017YFB0802704 and Program of Shanghai Technology Research Leader under Grant 16XD1424400.

References

- [1] M. Gruteser and D. Grunwald, "Anonymous usage of location-based services through spatial and temporal cloaking," in *Proceedings of the 1st International Conference on Mobile Systems, Applications and Services*, pp. 31–42, ACM, San Francisco, Calif, USA, 2003.
- [2] M. F. Mokbel, "Towards Privacy-Aware Location-Based database servers," in *Proceedings of the 22nd International Conference on Data Engineering Workshops, ICDEW 2006*, USA, April 2006.
- [3] P. Kalnis, G. Ghinita, K. Mouratidis, and D. Papadias, "Preventing location-based identity inference in anonymous spatial queries," *IEEE Transactions on Knowledge and Data Engineering*, vol. 19, no. 12, pp. 1719–1733, 2007.
- [4] D. Papadias, D. Zhang, and G. Kollios, *Advances in Spatial and Temporal Databases*, vol. 4605, Springer, Berlin, Germany, 2007.
- [5] C.-Y. Chow, M. F. Mokbel, and W. G. Aref, "Casper*: query processing for location services without compromising privacy," *ACM Transactions on Database Systems (TODS)*, vol. 34, no. 4, article 24, 2009.
- [6] J.-H. Um, H.-D. Kim, and J.-W. Chang, "An advanced cloaking algorithm using Hilbert curves for anonymous location based service," in *Proceedings of the 2nd International Conference on Social Computing*, pp. 1093–1098, Minneapolis, MN, USA, August 2010.
- [7] A.-A. Hossain, A. Hossain, H.-K. Yoo, and J.-W. Chang, "H-star: Hilbert-order based star network expansion cloaking algorithm in road networks," in *Proceedings of the 14th IEEE Int. Conf. on Computational Science and Engineering, CSE 2011, the 11th International Symposium on Pervasive Systems, Algorithms, and Networks, I-SPAN 2011, and the 10th IEEE Int. Conf. on Ubiquitous Computing and Communications, IUCC 2011*, pp. 81–88, chn, August 2011.
- [8] I.-T. Lien, Y.-H. Lin, J.-R. Shieh, and J.-L. Wu, "A novel privacy preserving location-based service protocol with secret circular shift for k-NN search," *IEEE Transactions on Information Forensics and Security*, vol. 8, no. 6, pp. 863–873, 2013.
- [9] Y. Utsunomiya, K. Toyoda, and I. Sasase, "LPCQP: Lightweight private circular query protocol with divided POI-table and somewhat homomorphic encryption for privacy-preserving k-NN search," *Journal of Information Processing*, vol. 24, no. 1, pp. 109–122, 2016.
- [10] Z. Brakerski, C. Gentry, and V. Vaikuntanathan, "(Leveled) fully homomorphic encryption without bootstrapping," in *Proceedings of the 3rd Conference on Innovations in Theoretical Computer Science, ITCS 2012*, pp. 309–325, usa, January 2012.
- [11] H. Sagan, *Space-Filling Curves*, Springer, New York, NY, USA, 1994.
- [12] D. Hilbert, "Ueber die stetige abbildung einer linie auf ein flächenstück," *Mathematische Annalen*, vol. 38, no. 3, pp. 459–460, 1891.
- [13] B. Moon, H. V. Jagadish, C. Faloutsos, and J. H. Saltz, "Analysis of the clustering properties of the Hilbert space-filling curve," *IEEE Transactions on Knowledge and Data Engineering*, vol. 13, no. 1, pp. 124–141, 2001.
- [14] H. V. Jagadish, "Linear clustering of objects with multiple attributes," in *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pp. 332–342, Atlantic City, NJ, USA, May 1990.
- [15] E. H. Moore, "On certain crinkly curves," *Transactions of the American Mathematical Society*, vol. 1, no. 1, pp. 72–90, 1900.
- [16] R. L. Rivest, L. Adleman, and M. L. Dertouzos, "On data banks and privacy homomorphisms," *Foundations of Secure Computation*, vol. 4, no. 11, pp. 169–180, 1978.
- [17] C. Gentry, "Fully Homomorphic Encryption Using Ideal Lattices," in *Proceedings of the 41st Annual ACM Symposium on Theory of Computing, STOC '09*, pp. 169–178, usa, June 2009.
- [18] M. van Dijk, C. Gentry, S. Halevi, and V. Vaikuntanathan, "Fully homomorphic encryption over the integers," in *Proceedings of the Annual International Conference on the Theory and Applications of Cryptographic Techniques (EUROCRYPT 2010)*, Lecture Notes in Comput. Sci., pp. 24–43, Springer.
- [19] N. P. Smart and F. Vercauteren, "Fully homomorphic encryption with relatively small key and ciphertext sizes," in *Proceedings of the International Workshop on Public Key Cryptography*, vol. 6056 of *Lecture Notes in Comput. Sci.*, pp. 420–443, Springer.
- [20] C. Gentry and S. Halevi, "Implementing Gentry's fully—homomorphic encryption scheme," in *Advances in cryptology—EUROCRYPT 2011*, vol. 6632 of *Lecture Notes in Computer Science*, pp. 129–148, Springer, Heidelberg, Germany, 2011.
- [21] C. Gentry and S. Halevi, "Fully homomorphic encryption without squashing using depth-3 arithmetic circuits," in *Proceedings of the 2011 IEEE 52nd Annual Symposium on Foundations of Computer Science, FOCS 2011*, pp. 107–116, 2011.
- [22] J.-S. Coron, D. Naccache, and M. Tibouchi, "Public key compression and modulus switching for fully homomorphic encryption over the integers," in *Proceedings of the Annual International Conference on the Theory and Applications of Cryptographic Techniques (EUROCRYPT 2012)*, vol. 7237 of *Lecture Notes in Comput. Sci.*, pp. 446–464, Springer.
- [23] S. Halevi and V. Shoup, "Design and implementation of a homomorphic-encryption library," *IBM Research*, pp. 12–15, 2013.
- [24] N. P. Smart and F. Vercauteren, "Fully homomorphic SIMD operations," *Designs, Codes and Cryptography*, vol. 71, no. 1, pp. 57–81, 2014.
- [25] C. Gentry, S. Halevi, and N. P. Smart, "Homomorphic evaluation of the AES circuit," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics): Preface*, vol. 7417, pp. 850–867, 2012.
- [26] J. Wu, S. Pan, X. Zhu, and Z. Cai, "Boosting for multi-graph classification," *IEEE Transactions on Cybernetics*, vol. 45, no. 3, pp. 430–443, 2015.
- [27] J. Wu, X. Zhu, C. Zhang, and P. S. Yu, "Bag Constrained Structure Pattern Mining for Multi-Graph Classification," *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, no. 10, pp. 2382–2396, 2014.
- [28] J. Wu, S. Pan, X. Zhu, C. Zhang, and X. Wu, "Positive and Unlabeled Multi-Graph Learning," *IEEE Transactions on Cybernetics*, vol. 47, no. 4, pp. 818–829, 2016.
- [29] J. Wu, S. Pan, X. Zhu, C. Zhang, and X. Wu, "Multi-instance learning with discriminative bag mapping," *IEEE Transactions on Knowledge and Data Engineering*, no. 99, p. 16, 2018.

- [30] J. Wu, S. Pan, X. Zhu, C. Zhang, and P. S. Yu, "Multiple Structure-View Learning for Graph Classification," *IEEE Transactions on Neural Networks and Learning Systems*, no. 99, pp. 1–16, 2017.
- [31] C. Bettini, S. Mascetti, X. S. Wang, D. Freni, and S. Jajodia, "Anonymity and historical-anonymity in location-based services," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics): Preface*, vol. 5599, pp. 1–30, 2009.
- [32] A. Machanavajjhala, D. Kifer, J. Gehrke, and M. Venkitasubramaniam, " ℓ -diversity: Privacy beyond k-anonymity," *ACM Transactions on Knowledge Discovery from Data (TKDD)*, vol. 1, no. 1, Article ID 1217302, 2007.
- [33] M. F. Mokbel, C. Y. Chow, and W. G. Aref, "The new Casper: query processing for location services without compromising privacy," in *Proceedings of the 32nd International Conference on Very Large Data Bases*, pp. 763–774, 2006.
- [34] R. Dewri, I. Ray, I. Ray, and D. Whitley, "On the formation of historically k-anonymous anonymity sets in a continuous LBS," *Lecture Notes of the Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering*, vol. 50, pp. 71–88, 2010.
- [35] A. Hossain, A.-A. Hossain, S.-J. Jang, Y.-S. Shin, and J.-W. Chang, "K-anonymous cloaking algorithm based on weighted adjacency graph for preserving location privacy," in *Proceedings of the 11th IEEE International Conference on Trust, Security and Privacy in Computing and Communications, TrustCom-2012*, pp. 358–365, UK, June 2012.
- [36] P. Shankar, Y.-W. Huang, P. Castro, B. Nath, and L. Iftode, "Crowds replace experts: Building better location-based services using mobile social network interactions," in *Proceedings of the 10th IEEE International Conference on Pervasive Computing and Communications, PerCom 2012*, pp. 20–29, Switzerland, March 2012.
- [37] Y. Wang, J. Peng, L.-P. He, T.-T. Zhang, and H.-Z. Li, "LBSs privacy preserving for continuous query based on semi-honest third parties," in *Proceedings of the IEEE 31st International Performance Computing and Communications Conference (IPCCC '12)*, pp. 384–391, IEEE, December 2012.
- [38] Y. Wang, H. Zhou, Y. Wu, and L. Sun, "Preserving location privacy for location-based services with continuous queries on road network," in *Proceedings of the 2012 7th International Conference on Computer Science and Education, ICCSE 2012*, pp. 822–827, Australia, July 2012.
- [39] H. Hu, J. Xu, Q. Chen, and Z. Yang, "Authenticating location-based services without compromising location privacy," in *Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data, SIGMOD '12*, pp. 301–312, May 2012.
- [40] L. Sweeney, "k-anonymity: A model for protecting privacy," *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 10, no. 5, pp. 557–570, 2002.
- [41] M. L. Yiu, C. S. Jensen, X. Huang, and H. Lu, "SpaceTwist: managing the trade-offs among location privacy, query performance, and query accuracy in mobile services," in *Proceedings of the IEEE 24th International Conference on Data Engineering (ICDE '08)*, pp. 366–375, IEEE Press, Cancun, Mexico, April 2008.
- [42] G. Ghinita, P. Kalnis, M. Kantarcioglu, and E. Bertino, "Approximate and exact hybrid algorithms for private nearest-neighbor queries with database protection," *GeoInformatica*, vol. 15, no. 4, pp. 699–726, 2011.
- [43] W. Ni, J. Zheng, and Z. Chong, "HilAnchor: location privacy protection in the presence of users' preferences," *Journal of Computer Science and Technology*, vol. 27, no. 2, pp. 413–427, 2012.
- [44] R. Shokri, P. Papadimitratos, G. Theodorakopoulos, and J.-P. Hubaux, "Collaborative location privacy," in *Proceedings of the 8th IEEE International Conference on Mobile Ad-hoc and Sensor Systems, MASS 2011*, pp. 500–509, Spain, October 2011.
- [45] A. Pingley, N. Zhang, and X. Fu, "Protection of query privacy for continuous location based services," in *Proceedings of the INFOCOM*, pp. 1710–1718, IEEE, 2011.
- [46] X. Chen and J. Pang, "Exploring dependency for query privacy protection in location-based services," in *Proceedings of the 3rd ACM Conference on Data and Application Security and Privacy, CODASPY 2013*, pp. 37–47, USA, February 2013.
- [47] X. Chen and J. Pang, "Measuring query privacy in location-based services," in *Proceedings of 2nd ACM International Conference on Data and Application Security and Privacy*, pp. 49–60, San Antonio, Tex, USA, February 2012.

Research Article

Privacy-Preserving and Scalable Service Recommendation Based on SimHash in a Distributed Cloud Environment

Yanwei Xu,¹ Lianyong Qi,¹ Wanchun Dou,² and Jiguo Yu¹

¹*School of Information Science and Engineering, Chinese Academy of Education Big Data, Qufu Normal University, Rizhao 276826, China*

²*State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210093, China*

Correspondence should be addressed to Lianyong Qi; lianyongqi@gmail.com

Received 4 August 2017; Revised 9 November 2017; Accepted 19 November 2017; Published 18 December 2017

Academic Editor: Chuan Zhou

Copyright © 2017 Yanwei Xu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

With the increasing volume of web services in the cloud environment, Collaborative Filtering- (CF-) based service recommendation has become one of the most effective techniques to alleviate the heavy burden on the service selection decisions of a target user. However, the service recommendation bases, that is, historical service usage data, are often distributed in different cloud platforms. Two challenges are present in such a cross-cloud service recommendation scenario. First, a cloud platform is often not willing to share its data to other cloud platforms due to privacy concerns, which decreases the feasibility of cross-cloud service recommendation severely. Second, the historical service usage data recorded in each cloud platform may update over time, which reduces the recommendation scalability significantly. In view of these two challenges, a novel privacy-preserving and scalable service recommendation approach based on SimHash, named SerRec_{SimHash}, is proposed in this paper. Finally, through a set of experiments deployed on a real distributed service quality dataset *WS-DREAM*, we validate the feasibility of our proposal in terms of recommendation accuracy and efficiency while guaranteeing privacy-preservation.

1. Introduction

With the ever-increasing volume and variety of web services in various web-based communities, it becomes a challenging task to find the web services that a target user is really interested in [1–3]. In this situation, various service recommendation techniques are introduced to alleviate the heavy burden on the service selection decisions of target users, for example, the well-adopted user-based Collaborative Filtering (i.e., UCF). According to traditional UCF, the similar friends of a target user are often employed to make recommendations to the target user [4]. Therefore, similar friend discovery is the key step to the subsequent service recommendation.

Generally, the bases for similar friend discovery, that is, historical service usage data (e.g., service quality observed by users) are centralized; in this situation, it is easy to determine the similar friends of a target user. However, in the age of IoT (Internet of Things), the quality data of various services are often monitored and collected by geographically distributed

sensors and stored in different cloud platforms [5]. In this situation, the historical service usage data are not centralized, but distributed. Such a distributed service recommendation scenario calls for data sharing and collaboration between different cloud platforms. However, as work [6] indicates, this kind of cross-platform data sharing may bring additional privacy leakage risk, which decreases the feasibility of cross-cloud service recommendation severely. Besides, for the involved multiple cloud platforms, their volume of service quality data may become increasingly huge with updates over time, which leads to a frequent recalculation of user similarity and hence reduces the recommendation scalability significantly.

In view of these two challenges, a novel privacy-preserving and scalable service recommendation approach based on SimHash, named SerRec_{SimHash}, is put forward in this paper. Our SerRec_{SimHash} can achieve a good recommendation performance in terms of accuracy, efficiency, and privacy-preservation.

Generally, the contributions of this paper are threefold:

- (1) To the best of our knowledge, existing research works seldom consider the service recommendation in a distributed cloud environment, as well as the resulting privacy-preservation problems. In this paper, we formalize this privacy-preserving service recommendation problem and clarify its research significance.
- (2) We put forward a novel service recommendation approach based on offline SimHash technique [7], named $\text{SerRec}_{\text{SimHash}}$, to protect the private information of most users in different cloud platforms, and meanwhile improve the service recommendation efficiency and scalability.
- (3) We conduct a set of experiments based on a real distributed service quality dataset $WS\text{-}DREAM$ to validate the feasibility of our proposed $\text{SerRec}_{\text{SimHash}}$ approach. Experiment results show that $\text{SerRec}_{\text{SimHash}}$ achieves a good performance in terms of recommendation accuracy and scalability while guaranteeing privacy-preservation.

The rest of the paper is organized as follows. Related work is presented in Section 2. Research motivation is demonstrated in Section 3. In Section 4, we introduce the details of our proposed service recommendation approach $\text{SerRec}_{\text{SimHash}}$. In Section 5, a set of experiments are conducted based on $WS\text{-}DREAM$ dataset, to validate the feasibility and advantages of our proposal. And finally, in Section 6, we summarize the paper and point out the future research directions.

2. Related Work

Collaborative Filtering (i.e., CF) has become one of the most effective techniques in various recommender systems. User-based CF and item-based CF are brought forth for high-quality service recommendation in [4] and [8], respectively. In order to combine their advantages, a hybrid CF recommendation approach is introduced in [9]. Experiment results show that the hybrid approach improves the recommendation performance. As the quality of a web service often depends on the service execution context (e.g., time, location), time-aware CF and location-aware CF are proposed in [10] and [11], respectively, to improve the accuracy of recommended results. However, the above approaches cannot handle the recommendation problems where historical service usage data are very sparse. In view of this drawback, a belief propagation-based approach is proposed in [12], to find the potential friends of the target user.

However, the above approaches all assume that the service recommendation bases, that is, historical service usage data, are centralized, without considering the distributed service recommendation scenarios as well as the resulting privacy leakage risk. In view of this drawback, the authors in [13] suggest that a user should release only a small portion of his/her observed service quality data to the public so that the remaining majority of user-service quality data are secure. However, the released small portion of data can still reveal

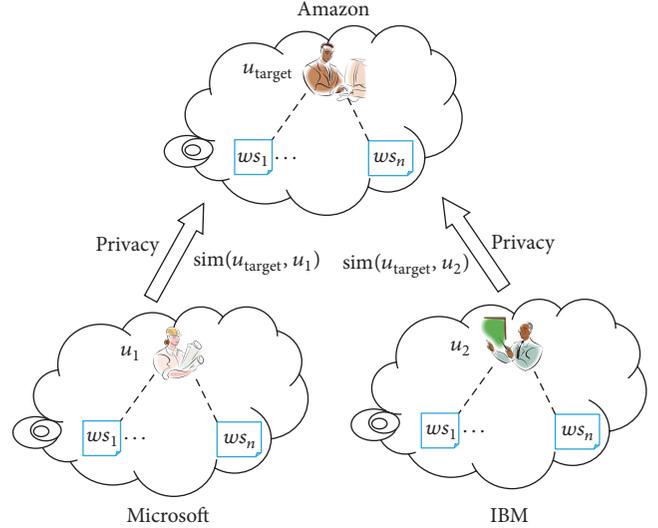


FIGURE 1: Cross-cloud service recommendation: an example.

part of a user's private information. In order to protect user privacy completely, the data obfuscation technique is adopted in [14] to hide the real service quality data by adding an obfuscated data item. However, as the service quality data used to make service recommendations have been obfuscated, the recommendation accuracy is decreased accordingly; besides, additional time cost is brought by the adopted data obfuscation operation. Similarly, a segment-based data hiding approach is introduced in [15], where each piece of user-service quality data is divided into several data segments, and then the data segments are employed to calculate user similarity approximately and make further service recommendation. However, there are still two shortcomings in this approach. First, the data segmentation process often takes much time, which decreases the recommendation efficiency heavily. Second, it fails to protect some important privacy information appropriately, for example, the information of the service intersection commonly invoked by two users. Locality-sensitive hashing technique is recruited in [16] to protect and realize the privacy-preservation purpose; however, only partial private information of users can be protected very well.

In view of the drawbacks of existing approaches, a novel privacy-preserving and scalable service recommendation approach based on SimHash, that is, $\text{SerRec}_{\text{SimHash}}$, is proposed in this paper, to cope with the service recommendation problems in the distributed cloud environment. Next, an example is presented in Section 3 to further demonstrate the research motivation of our paper.

3. Research Motivation

An intuitive example is presented in Figure 1 to motivate our paper. Here, u_{target} denotes a target user to whom *Amazon* platform intends to recommend services; u_1 and u_2 are two users whose observed service quality data are recorded in *Microsoft* and *IBM* platforms, respectively; $\{ws_1, \dots, ws_n\}$ are the candidate services for recommendation. Specifically, if a

Step 1 (building user indexes offline based on SimHash). For each user $u_i \in U$, calculate his/her hash value $H(u_i)$ offline based on SimHash. Then $H(u_i)$ is regarded as the index for u_i .

Step 2 (finding “probably similar” friends of the target user). According to the same hash function adopted in Step 1, calculate user index for u_{target} , that is, $H(u_{\text{target}})$. If the Hamming Distance between $H(u_{\text{target}})$ and $H(u_i)$ is smaller than 3, then u_i is considered as a “probably similar” friend of u_{target} .

Step 3 (finding “really similar” friends of the target user). For a “probably similar” friend u_i obtained in Step 2, calculate his/her similarity with u_{target} ; if the similarity is larger than a threshold P , then u_i is a “really similar” friend of u_{target} .

Step 4 (service recommendation). According to u_{target} ’s “really similar” friends derived in Step 3, predict the quality of services never invoked by u_{target} and recommend the quality-optimal services to u_{target} .

Box 1: Four steps of service recommendation approach SerRec_{SimHash}.

user has never invoked a service, the corresponding service quality data is null.

Next, according to traditional UCF, the first step is to calculate user similarity $\text{sim}(u_{\text{target}}, u_1)$ and $\text{sim}(u_{\text{target}}, u_2)$ so as to determine the similar friends of u_{target} . However, the above user similarity calculation process involves the cross-platform collaborations and hence faces the following two challenges:

- (1) Generally, *Microsoft* and *IBM* are not willing to share their recorded service quality data to *Amazon* due to privacy concerns, which decreases the feasibility of cross-cloud user similarity calculation and subsequent service recommendation severely.
- (2) In *Amazon*, *Microsoft*, and *IBM*, the volume of service quality data may become increasingly huge with updates over time; in this situation, the collaboration efficiency and scalability are often reduced significantly and hence cannot satisfy the quick recommendation requirements from target users.

In view of these two challenges, a privacy-preserving and scalable service recommendation approach, that is, SerRec_{SimHash}, is proposed in this paper, which will be introduced in detail in the next section.

4. A SimHash-Based Service Recommendation Approach

In this section, a privacy-preserving and scalable approach, that is, SerRec_{SimHash}, is proposed to handle the distributed service recommendation problems. The main idea behind SerRec_{SimHash} is: the users who have invoked the most common services can be regarded as “probably similar” friends [17]; therefore, we first utilize SimHash to look for a small number of “probably similar” friends of the target user, in a privacy-preserving and scalable way; afterwards, we determine the target user’s “really similar” friends from the “probably similar” ones; finally, we make service recommendations to the target user based on the preferences of his/her “really similar” friends.

Concretely, SerRec_{SimHash} consists of the four steps in Box 1. Here, u_{target} denotes a target user, U is the user set in multiple involved cloud platforms, $\{ws_1, \dots, ws_n\}$ is the candidate service set, and $H(u)$ denotes the hash value of user u based on SimHash.

Step 1 (building user indexes offline based on SimHash). For each user $u_i \in U$, according to his/her historical service invocation records, we can build his/her index offline, denoted by $H(u_i)$, based on SimHash technique (see Figure 2). Here, m and n denote the number of users and number of services, respectively. Next, we introduce how to obtain $H(u_i)$.

First, for each service $ws_j \in WS$, we can generate a random r -dimensional 0-1 vector v_j where $r = \lceil \log_2^n \rceil$ (here, $\lceil x \rceil$ means the upper integer of x ; e.g., $\lceil 3.4 \rceil = 4$). Considering the example in Figure 2, $r = 6$ and the 0-1 vector corresponding to service ws_1 , that is, $v_1 = (0, 0, 0, 0, 0, 1)$ holds. Then according to the historical service invocation records, u_i can be denoted by a n -dimensional vector $h_1(u_i) = (V_1, \dots, V_n)$ in (1).

$$V_j = \begin{cases} v_j & \text{if } u_i \text{ has invoked } ws_j \text{ before} \\ \text{Null} & \text{if } u_i \text{ has never invoked } ws_j \text{ before.} \end{cases} \quad (1)$$

Next, in vector $h_1(u_i)$, we drop the dimensions with null value and replace value “0” by value “-1”, after which a new vector $h_2(u_i)$ is achieved (see Figure 2(2)). Then for the derived $n * r$ (at most) matrix corresponding to vector $h_2(u_i)$, we calculate the sum of its each column. Afterwards, we obtain a new vector $h_3(u_i)$ (see Figure 2(3)), where the positive and negative values are replaced by “1” and “0”, respectively, after which r -dimensional 0-1 vector $H(u_i)$ (see Figure 2(4)) is obtained. Then according to SimHash theory [6], $H(u_i)$ can be regarded as the index for user u_i . This way, we can build indexes for all the users in set U .

For a user, his/her historical service invocation data are recorded by a certain cloud platform (e.g., *Amazon* or *Microsoft* or *IBM* in Figure 1); therefore, the user index can be built offline beforehand by the cloud platform so as to

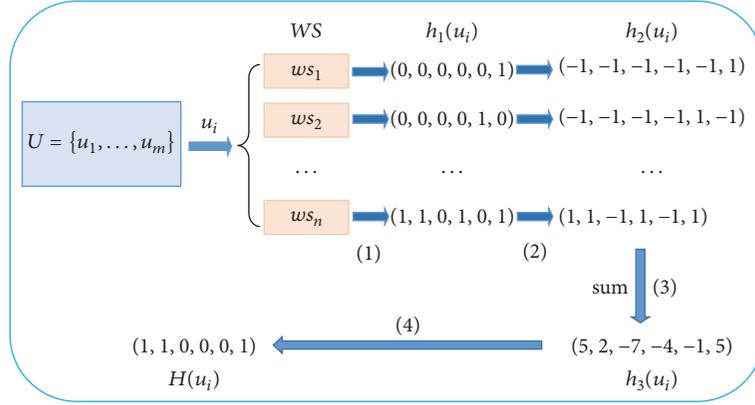


FIGURE 2: User indexes building procedure in Step 1: an example.

reduce the time cost. Besides, through SimHash, each user u_i is encapsulated into a less-sensitive user index $H(u_i)$, without revealing his/her sensitive information (e.g., *whether he/she has invoked a service or not, a service's running quality observed by him/her*) to other platforms. Therefore, user privacy is protected.

Step 2 (finding “probably similar” friends of the target user). According to same hash functions adopted in Step 1, we calculate the index for the target user, that is, $H(u_{\text{target}})$. Next, we calculate the Hamming Distance between $H(u_{\text{target}})$ and $H(u_i)$ ($1 \leq i \leq m$), denoted by $D(H(u_{\text{target}}), H(u_i))$. Concretely, suppose $H(u_{\text{target}})$ and $H(u_i)$ are denoted by r -dimensional vectors $(v_{\text{target-1}}, \dots, v_{\text{target-r}})$ and $(v_{i-1}, \dots, v_{i-r})$, respectively. Then $D(H(u_{\text{target}}), H(u_i))$ can be calculated by (2), where a_k is a Boolean value calculated by (3). Here, symbol “ \oplus ” denotes the XOR operation.

$$D(H(u_{\text{target}}), H(u_i)) = \sum a_k \quad (1 \leq k \leq r) \quad (2)$$

$$a_k = v_{\text{target-k}} \oplus v_{i-k} = \begin{cases} 1 & \text{if } v_{\text{target-k}} \neq v_{i-k} \\ 0 & \text{if } v_{\text{target-k}} = v_{i-k} \end{cases} \quad (3)$$

According to SimHash [6], if $D(H(u_{\text{target}}), H(u_i)) < 3$ holds, then we can conclude that the services invoked by u_{target} and u_i are approximately the same. In other words, u_i can be regarded as a “probably similar” friend of u_{target} and then put into set $\text{Prob_Sim_Friend}(u_{\text{target}})$. Moreover, the size of $\text{Prob_Sim_Friend}(u_{\text{target}})$, that is, $|\text{Prob_Sim_Friend}(u_{\text{target}})|$, is often small ($\ll m$) due to the nature of SimHash.

Step 3 (finding “really similar” friends of the target user). The users in set $\text{Prob_Sim_Friend}(u_{\text{target}})$ (obtained in Step 2) are only “probably similar” friends of the target user, not necessarily “really similar” friends. Considering this point, in this step, we further determine the “really similar” friends of the target user from set $\text{Prob_Sim_Friend}(u_{\text{target}})$. Concretely, for any $u_i \in \text{Prob_Sim_Friend}(u_{\text{target}})$, we calculate his/her similarity with u_{target} , that is, $\text{Sim}(u_{\text{target}}, u_i)$, based on Pearson Correlation Coefficient (PCC) [18] in (4) (as

$|\text{Prob_Sim_Friend}(u_{\text{target}})|$ is often small, only a small number of users take part in the user similarity calculation process in (4); as a consequence, we can protect the private service quality data observed by the remaining majority of users).

In (4), symbol I denotes the service intersection invoked by u_{target} and u_i ; q is a quality dimension of web services, for example, *response time*; $q_{\text{target-j}}$ and q_{i-j} represent service ws_j 's quality values over dimension q observed by u_{target} and u_i , respectively; \bar{q}_{target} and \bar{q}_i denote u_{target} 's and u_i 's average quality values over dimension q of all the services invoked by u_{target} and u_i , respectively. Specifically, if the service intersection $I = \text{Null}$, $\text{Sim}(u_{\text{target}}, u_i) = 0$ holds. Moreover, if condition in (5) holds, u_i can be regarded as a “really similar” friend of u_{target} and put into set $\text{Real_Sim_Friend}(u_{\text{target}})$. Here, symbol P is a predefined similarity threshold ($0.5 \leq P \leq 1$).

$$\begin{aligned} \text{Sim}(u_{\text{target}}, u_i) &= \frac{\sum_{ws_j \in I} (q_{\text{target-j}} - \bar{q}_{\text{target}}) * (q_{i-j} - \bar{q}_i)}{\sqrt{\sum_{ws_j \in I} (q_{\text{target-j}} - \bar{q}_{\text{target}})^2} * \sqrt{\sum_{ws_j \in I} (q_{i-j} - \bar{q}_i)^2}} \quad (4) \end{aligned}$$

$$\text{Sim}(u_{\text{target}}, u_i) \geq P. \quad (5)$$

Step 4 (service recommendation). For all the users u_i in set $\text{Real_Sim_Friend}(u_{\text{target}})$ (obtained in Step 3), we rank them by $\text{Sim}(u_{\text{target}}, u_i)$ (see (4)) in descending order and return the Top 3 (at most) similar friends (denoted by set $U_{\text{top-3}}$) of the target user. Afterwards, for each service never invoked by the target user, denoted by ws_j , we predict its quality over dimension q observed by u_{target} , that is, $q_{\text{target-j}}$, by (6), where $u_i \in U_{\text{top-3}}$ and q_{i-j} represents service ws_j 's quality value over dimension q observed by u_i . Finally, we select the service with the optimal predicted quality and recommend it to the target user, so as to finish the whole service recommendation process.

$$q_{\text{target-j}} = \frac{\sum_{u_i \in U_{\text{top-3}}} \text{Sim}(u_{\text{target}}, u_i) * q_{i-j}}{\sum_{u_i \in U_{\text{top-3}}} \text{Sim}(u_{\text{target}}, u_i)}. \quad (6)$$

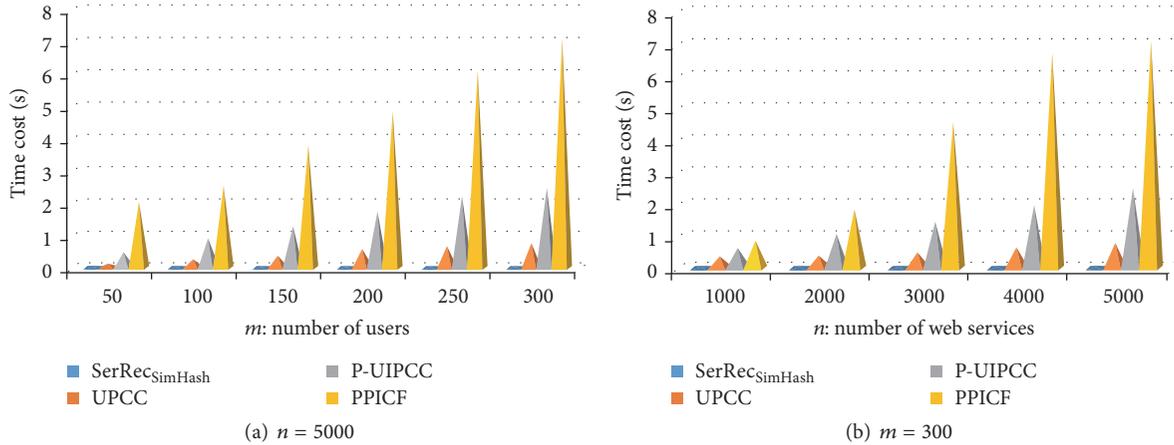


FIGURE 3: Recommendation efficiency comparison of four approaches.

5. Experiments

5.1. Experiment Configurations. In this section, a set of experiments are deployed on *WS-DREAM* dataset [19] to validate the feasibility of our proposed recommendation approach SerRec_{SimHash}. *WS-DREAM* is a real-world service quality (e.g., *throughput*) set obtained from 339 users on 5825 web services from different countries. To simulate the recommendation scenario that we focus on in this paper (i.e., recommendation in a distributed cloud environment), each country is regarded as a cloud platform.

We compare our approach with a benchmark approach UPCC [20] and another two up-to-date privacy-preserving recommendation approaches, that is, P-UIPCC [14] and PPICF [15]. Many works, for example, [21–23], consider the *time cost* and the *MAE* as the evaluation criteria; likewise, we also adopt these two criteria in this paper (in our SerRec_{SimHash} approach, most user privacy information, e.g., *whether a user has invoked a service or not* and *service quality observed by a user*, can be protected by the intrinsic nature of SimHash; therefore, we will not evaluate the capability of privacy-preservation of our proposal here).

- (1) *Time cost*: the consumed time for recommending a web service to the target user, which can be used to measure the recommendation efficiency and scalability.
- (2) *MAE*: the difference between the predicted quality and real quality of recommended services (the smaller the better), which can be used to measure the recommendation accuracy.

The density of user-service quality matrix is set at 3% and the experiments are conducted on a Lenovo laptop with 2.40 GHz processor and 12.0 GB RAM. The laptop is running under Windows 10 and JAVA 8. Each experiment is repeated 10 times and the average experiment results are reported.

5.2. Experiment Results and Analyses. Concretely, the following four profiles are tested and compared, respectively. Here,

m and n denote the number of users and number of web services, respectively; user similarity threshold $P = 0.5$ holds.

Profile 1: Recommendation Efficiency Comparison. In this profile, we test the time cost of our proposal with respect to m and n and compare it with the remaining three approaches. The experiment parameters are set as follows: m is varied from 50 to 300; n is varied from 1000 to 5000. The concrete experiment results are shown in Figure 3 ($n = 5000$ holds in Figure 3(a) and $m = 300$ holds in Figure 3(b)).

As can be seen from Figure 3(a), the time costs of UPCC, P-UIPCC, and PPICF approaches all increase approximately linearly with the growth of m ; this is because more time is needed to calculate user similarities when the number of users, that is, m , becomes larger, while our proposed SerRec_{SimHash} approach outperforms those three ones in terms of time cost, as most jobs (e.g., user indexes building) can be finished offline before a service recommendation request arrives. Furthermore, after the hashing process, only a few “probably similar” friends of the target user are obtained; as a consequence, little time is taken to find the “really similar” friends of the target user from the small number of “probably similar” friends. Due to the above two reasons, the recommendation efficiency and scalability of our proposed SerRec_{SimHash} approach are improved significantly. Similar comparison results can be observed from Figure 3(b), whose reasons are the same as those in Figure 3(a) and will not be discussed repeatedly.

Profile 2: Recommendation Accuracy Comparison. Accuracy is a key criterion to evaluate the quality of a recommender system. Therefore, in this profile, we test the *MAE* (the smaller the better) of our proposal and compare it with the remaining three approaches. The experiment parameters are set as follows: m is varied from 50 to 300; n is varied from 1000 to 5000. The experiment results are presented in Figure 4 ($n = 5000$ holds in Figure 4(a) and $m = 150$ holds in Figure 4(b)).

As Figure 4 shows, the recommendation accuracy values of P-UIPCC and PPICF approaches are often low (i.e., *MAE* values are high), as many approximate operations are

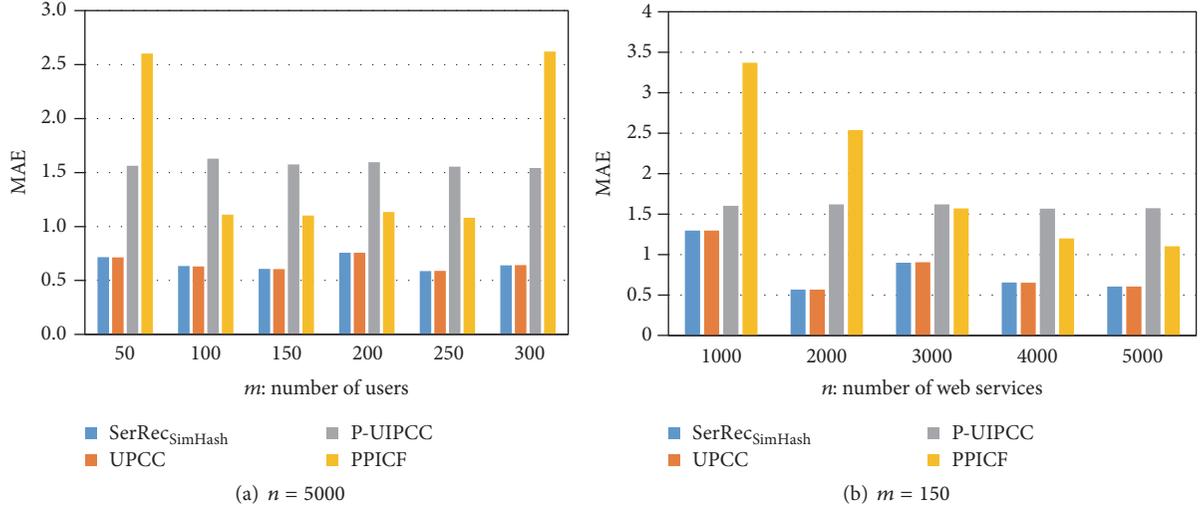
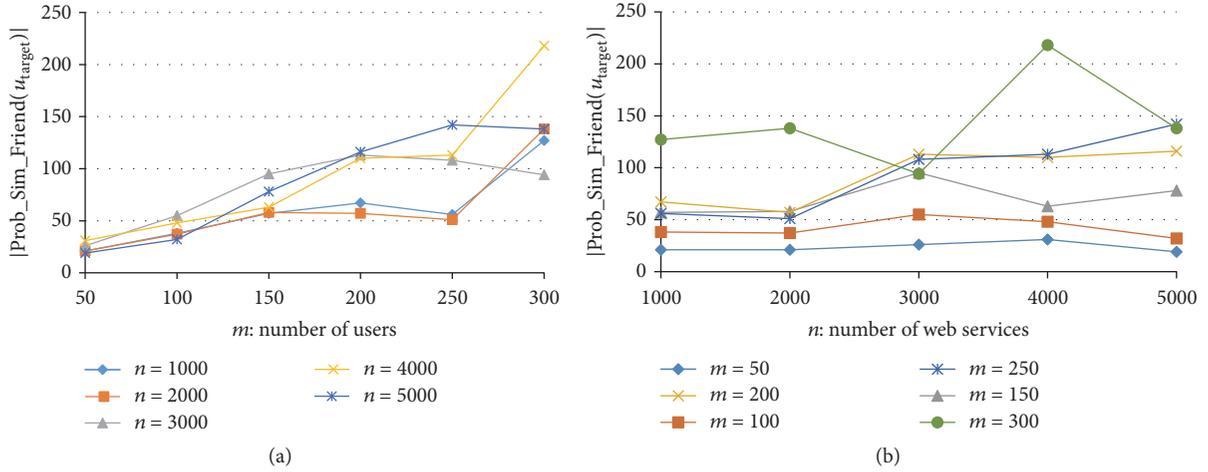


FIGURE 4: Recommendation accuracy comparison of four approaches.

FIGURE 5: $|\text{Prob_Sim_Friend}(u_{\text{target}})|$ with respect to m and n .

recruited in these two approaches to protect the user privacy, for example, data obfuscation technique adopted in *P-UIPCC* approach and data segmentation-merging technique recruited in *PPICF* approach. These techniques on one hand can protect the privacy information of users effectively and on the other hand decrease the accuracy of recommended results, while our proposed *SerRec_{SimHash}* approach achieves the approximate service recommendation accuracy as the benchmark approach *UPCC*, as the *SimHash* technique adopted in *SerRec_{SimHash}* can guarantee finding the “really similar” friends of a target user with high probability and thereby can achieve a high recommendation accuracy.

Profile 3: Number of “Probably Similar” Friends of the Target User in SerRec_{SimHash} with respect to m and n . In our *SerRec_{SimHash}* approach, a small number of “probably similar” friends (the number is $|\text{Prob_Sim_Friend}(u_{\text{target}})|$) of a target user are obtained. In this profile, we test the relationship

between $|\text{Prob_Sim_Friend}(u_{\text{target}})|$ and m and n . Experiment parameters are set as follows: m is varied from 50 to 300; n is varied from 1000 to 5000. The concrete experiment results are presented in Figure 5.

As Figure 5(a) shows, the value of $|\text{Prob_Sim_Friend}(u_{\text{target}})|$ increases approximately linearly with the growth of m ; this is because it is more probable to find a “probable friend” of the target user when the candidate user space becomes larger. As Figure 5(b) shows, the value of $|\text{Prob_Sim_Friend}(u_{\text{target}})|$ increases relatively slowly when n rises, whose reasons are twofold. First, more valuable recommendation information is available when the number of services, that is, n , increases; as a consequence, more “probably similar” friends of the target user can be found by our proposed *SerRec_{SimHash}* approach. Second, due to the intrinsic nature of *SimHash* technique adopted in our *SerRec_{SimHash}* approach, the number of services, that is, n , does not influence the finding process of “probably similar”

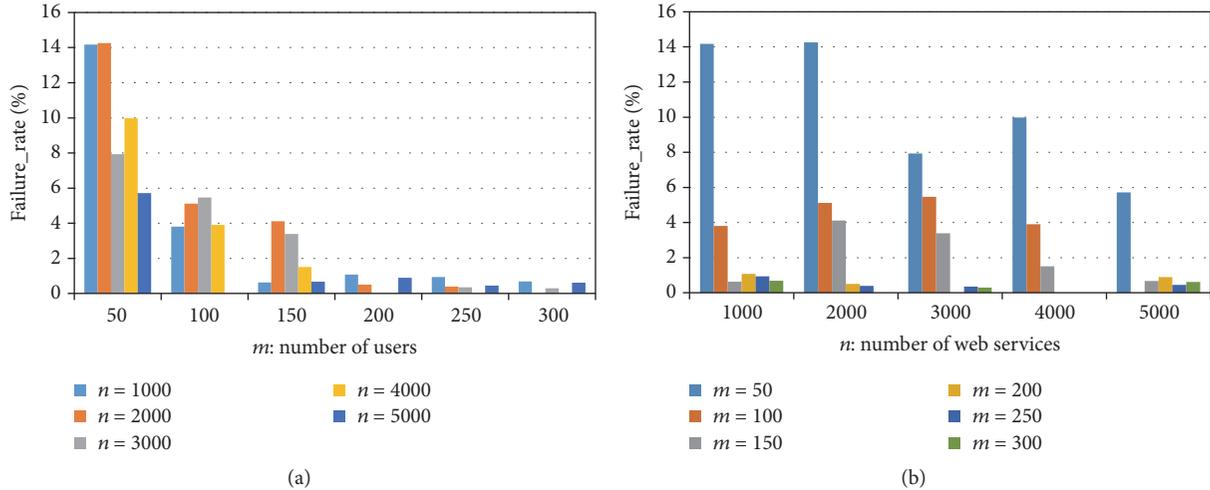


FIGURE 6: Recommendation failure rate of SerRec_{SimHash} with respect to m and n .

friends directly in our proposal and, hence, the influence of parameter n stressed on $|\text{Prob_Sim_Friend}(u_{\text{target}})|$ is not so obvious as that in Figure 5(a).

Profile 4: Recommendation Failure Rate of SerRec_{SimHash} with respect to m and n . The SimHash technique adopted in this paper is essentially a kind of probability-based similar neighbor finding approach [24]. Therefore, our proposed SerRec_{SimHash} approach may fail to return any recommended result in certain situations, that is, a failure occurs. Considering this point, in this profile, we test the recommendation failure rate of SerRec_{SimHash} with respect to m and n . Concretely, failure rate can be measured by the equation in (7), where Num_{success} and Num_{fail} represent the number of successful service recommendations and the number of failed service recommendations, respectively. The concrete experiment parameters are set as follows: m is varied from 50 to 300; n is varied from 1000 to 5000. The experiment results are shown in Figure 6.

$$\text{failure_rate} = \left(\frac{\text{Num}_{\text{fail}}}{\text{Num}_{\text{success}} + \text{Num}_{\text{fail}}} \right) * 100\%. \quad (7)$$

As Figure 6(a) shows, the failure rate of SerRec_{SimHash} approach decreases with the growth of m ; this is because it is more probable to find the “probably similar” friends of a target user when the candidate space of users becomes larger. Moreover, the failure rate approaches 0 when m is large enough, for example, when $m = 200, 250, \text{ or } 300$. Figure 6(b) shows the relationship between failure rate of SerRec_{SimHash} and the number of services, that is, n . As indicated in Figure 6(b), the failure rate approximately drops with the growth of n ; this is because when the number of services increases, the probability that two users have invoked the common services grows accordingly, and hence it is more probable to find the “probably similar” friends of a target user. Furthermore, as can be seen from Figure 6(b), the failure rate of SerRec_{SimHash} approach approaches 0 when n is large enough, for example, when $n = 5000$.

5.3. Shortcoming Analyses. In terms of the experiment results, we can conclude that SerRec_{SimHash} approach achieves a good tradeoff among the recommendation accuracy, efficiency, and failure rate while guaranteeing privacy-preservation. However, other evaluation criteria are not discussed in depth, such as the well-known consistency criterion (e.g., the inferred friend consistency) suggested in work [25]. Besides, as [26] indicates, weight plays an important role in the final evaluation results; however, we do not consider the weight of found friends in this paper for simplicity.

6. Conclusions and Future Work

In the distributed cloud environment, a cloud platform is often not willing to share its recorded user-service invocation data with other cloud platforms due to privacy concerns, which decreases the feasibility of cross-cloud collaborative service recommendation severely. Besides, the user-service invocation data recorded by each cloud platform may update over time, which reduces the recommendation scalability significantly. In view of these two challenges, a novel privacy-preserving and scalable service recommendation approach based on SimHash, that is, SerRec_{SimHash}, is put forward in this paper. To validate the feasibility of our proposal, we conduct a set of experiments based on a real distributed service quality dataset *WS-DREAM*. Experiment results show that SerRec_{SimHash} outperforms the other up-to-date approaches in terms of recommendation accuracy and efficiency while guaranteeing privacy-preservation.

As work [27] indicates, SimHash is essentially a probability-based search technique and, hence, failure is inevitable in certain situations. Considering this point, in the future, we will continue to refine our proposal so as to further decrease the recommendation failure rate and boost the recommendation robustness. Besides, due to the inherent shortcoming of various hash-based privacy-preservation techniques suggested in [28], it is hard to evaluate the privacy-preservation performance of our proposal. In the future, we hope to find well-adopted technical criteria

to evaluate the effectiveness of our proposal in terms of privacy-preservation. Moreover, work [29] proposes to utilize the semantic information to improve the retrieval performance; likewise, we hope to refine our work by adding more semantic information in the future.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This paper is partially supported by the Natural Science Foundation of China (no. 61402258, no. 61672276, no. 61373027, and no. 61672321), Key Research and Development Project of Jiangsu Province (no. BE2015154, no. BE2016120), and Open Project of State Key Laboratory for Novel Software Technology (no. KFKT2016B22).

References

- [1] Y. Xia, M. Zhou, X. Luo, Q. Zhu, J. Li, and Y. Huang, "Stochastic modeling and quality evaluation of infrastructure-as-a-service clouds," *IEEE Transactions on Automation Science and Engineering*, vol. 12, no. 1, pp. 162–170, 2015.
- [2] N. Zhang, J. Wang, and Y. Ma, "Mining domain knowledge on service goals from textual service descriptions," *IEEE Transactions on Services Computing*, no. 99, article 1, 2017.
- [3] J. Wang, Z. Zhu, J. Liu, C. Wang, and Y. Xu, "An approach of role updating in context-aware role mining," *International Journal of Web Services Research*, vol. 14, no. 2, pp. 24–44, 2017.
- [4] H. Rong, S. Huo, C. Hu, and J. Mo, "User similarity-based collaborative filtering recommendation algorithm," *Journal on Communication*, vol. 35, no. 2, pp. 16–24, 2014.
- [5] L. Qi, H. Xiang, W. Dou, C. Yang, Y. Qin, and X. Zhang, "Privacy-preserving distributed service recommendation based on locality-sensitive hashing," in *Proceedings of the 2017 IEEE International Conference on Web Services (ICWS)*, pp. 49–56, Honolulu, Hawaii, USA, June 2017.
- [6] J. Shen, T. Zhou, D. He, Y. Zhang, X. Sun, and Y. Xiang, "Block design-based key agreement for group data sharing in cloud computing," *IEEE Transactions on Dependable and Secure Computing*, no. 99, article 1, 2017.
- [7] Q. Jiang and M. Sun, "Semi-supervised SimHash for efficient document similarity search," in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, vol. 1, pp. 93–101, 2011.
- [8] K.-Y. Chung, D. Lee, and K. J. Kim, "Categorization for grouping associative items using data mining in item-based collaborative filtering," *Multimedia Tools and Applications*, vol. 71, no. 2, pp. 889–904, 2014.
- [9] C. Jiang, R. Duan, H. K. Jain, S. Liu, and K. Liang, "Hybrid collaborative filtering for high-involvement products: a solution to opinion sparsity and dynamics," *Decision Support Systems*, vol. 79, pp. 195–208, 2015.
- [10] X. Wang, J. Zhu, Z. Zheng, W. Song, Y. Shen, and M. R. Lyu, "A spatial-temporal qoS prediction approach for time-aware web service recommendation," *ACM Transactions on the Web (TWEB)*, vol. 10, no. 1, article 7, 2016.
- [11] C. Yu and L. Huang, "A Web service QoS prediction approach based on time- and location-aware collaborative filtering," *Service Oriented Computing and Applications*, vol. 10, no. 2, pp. 135–149, 2016.
- [12] L. Qi, X. Xu, X. Zhang et al., "Structural balance theory-based e-commerce recommendation over big rating data," *IEEE Transactions on Big Data*, no. 99, article 1, 2016.
- [13] W. Dou, X. Zhang, J. Liu, and J. Chen, "HireSome-II: towards privacy-aware cross-cloud service composition for big data applications," *IEEE Transactions on Parallel and Distributed Systems*, vol. 26, no. 2, pp. 455–466, 2015.
- [14] J. Zhu, P. He, Z. Zheng, and M. R. Lyu, "A privacy-preserving qos prediction framework for web service recommendation," in *Proceedings of the IEEE International Conference on Web Services (ICWS '15)*, pp. 241–248, New York, NY, USA, July 2015.
- [15] D. Li, C. Chen, Q. Lv et al., "An algorithm for efficient privacy-preserving item-based collaborative filtering," *Future Generation Computer Systems*, vol. 55, pp. 311–320, 2016.
- [16] Z. Xia, X. Wang, L. Zhang, Z. Qin, X. Sun, and K. Ren, "A privacy-preserving and copy-deterrence content-based image retrieval scheme in cloud computing," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 11, pp. 2594–2608, 2016.
- [17] Z. Zheng, H. Ma, M. R. Lyu, and I. King, "QoS-aware web service recommendation by collaborative filtering," *IEEE Transactions on Services Computing*, vol. 4, no. 2, pp. 140–152, 2011.
- [18] J. L. Rodgers and A. W. Nicewander, "Thirteen ways to look at the correlation coefficient," *The American Statistician*, vol. 42, no. 1, pp. 59–66, 1988.
- [19] Z. Zheng, Y. Zhang, and M. R. Lyu, "Investigating QoS of real-world web services," *IEEE Transactions on Services Computing*, vol. 7, no. 1, pp. 32–39, 2014.
- [20] J. S. Breese, D. Heckerman, and C. Kadie, "Empirical analysis of predictive algorithms for collaborative filtering," in *Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence (UAI '98)*, pp. 43–52, Madison, Wis, USA, July 1998.
- [21] Y. Xia, M. Zhou, X. Luo, S. Pang, and Q. Zhu, "Stochastic modeling and performance analysis of migration-enabled and error-prone clouds," *IEEE Transactions on Industrial Informatics*, vol. 11, no. 2, pp. 495–504, 2015.
- [22] W. Zheng, M. Zhou, L. Wu et al., "Percentile performance estimation of unreliable iaas clouds and their cost-optimal capacity decision," *IEEE Access*, vol. 5, pp. 2808–2818, 2017.
- [23] Z. Zhou, Q. J. Wu, and X. Sun, "Encoding multiple contextual clues for partial-duplicate image retrieval," *Pattern Recognition Letters*.
- [24] S. Sood and D. Loguinov, "Probabilistic near-duplicate detection using simhash," in *Proceedings of the 20th ACM Conference on Information and Knowledge Management (CIKM '11)*, pp. 1117–1126, Glasgow, UK, October 2011.
- [25] Z. Zhou, Y. Wang, Q. M. J. Wu, C.-N. Yang, and X. Sun, "Effective and efficient global context verification for image copy detection," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 1, pp. 48–63, 2017.
- [26] Z. Fu, X. Wu, Q. Wang, and K. Ren, "Enabling central keyword-based semantic extension search over encrypted outsourced data," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 12, pp. 2986–2997, 2017.
- [27] Z. Fu, J. Shu, J. Wang, Y. Liu, and S. Lee, "Privacy-preserving smart similarity search based on simhash over encrypted data in cloud computing," *Journal of Internet Technology*, vol. 16, no. 3, pp. 453–460, 2015.

- [28] Z. Zhou, Q. M. Jonathan Wu, C. Yang, X. Sun, and Z. Pan, "Coverless image steganography based on histograms of oriented gradients-based hashing algorithm," *Journal of Internet Technology*, vol. 18, no. 5, pp. 1177–1184, 2017.
- [29] Z. Fu, F. Huang, K. Ren, J. Weng, and C. Wang, "Privacy-preserving smart semantic search based on conceptual graphs over encrypted outsourced data," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 8, pp. 1874–1884, 2017.

Research Article

Kernel Negative ε Dragging Linear Regression for Pattern Classification

Yali Peng,^{1,2,3} Lu Zhang,^{1,2} Shigang Liu,^{1,2,3} Xili Wang,^{1,2} and Min Guo^{2,3}

¹Key Laboratory of Modern Teaching Technology, Ministry of Education, Xi'an 710062, China

²Engineering Laboratory of Teaching Information Technology of Shaanxi Province, Xi'an 710119, China

³School of Computer Science, Shaanxi Normal University, Xi'an 710119, China

Correspondence should be addressed to Shigang Liu; shgliu@snnu.edu.cn

Received 27 August 2017; Accepted 9 November 2017; Published 10 December 2017

Academic Editor: Chuan Zhou

Copyright © 2017 Yali Peng et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Linear regression (LR) and its variants have been widely used for classification problems. However, they usually predefine a strict binary label matrix which has no freedom to fit the samples. In addition, they cannot deal with complex real-world applications such as the case of face recognition where samples may not be linearly separable owing to varying poses, expressions, and illumination conditions. Therefore, in this paper, we propose the kernel negative ε dragging linear regression (KNDLR) method for robust classification on noised and nonlinear data. First, a technique called negative ε dragging is introduced for relaxing class labels and is integrated into the LR model for classification to properly treat the class margin of conventional linear regressions for obtaining robust result. Then, the data is implicitly mapped into a high dimensional kernel space by using the nonlinear mapping determined by a kernel function to make the data more linearly separable. Finally, our obtained KNDLR method is able to partially alleviate the problem of overfitting and can perform classification well for noised and deformable data. Experimental results show that the KNDLR classification algorithm obtains greater generalization performance and leads to better robust classification decision.

1. Introduction

Least squares regression (LSR) has been widely used for many fields of pattern recognition and computer vision. Owing to LSR being mathematically tractable and computationally efficient, in the past, many variants have been proposed. Notable LSR algorithms include weighted LSR [1], partial LSR [2], and other extensions (e.g., nonnegative least squares (NNLS) [3]). In the pattern recognition community, LSR is also referred to as minimum squared error algorithm [4–6]. Moreover, very competent extensions of least squares regression such as regularized least squares regression [7] are also proposed. Among extensions of least squares regression, sparse regression [8] and low-rank regression [9, 10] can obtain notable performance. The relationship between the regression and other methods such as locally linear embedding and local tangent space alignment is also studied [11]. In addition, LSR is also applied to semisupervised learning. Nie et al. [12] proposed adaptive loss minimization for semisupervised elastic embedding. Fang et al. [13] proposed

learning a nonnegative sparse graph for linear regression for semisupervised learning, in which linear regression and graph learning were simultaneously performed to guarantee an overall optimum.

LSR can be simply described as follows. Before conventional least squares regression (CLSR) is applied for classification [3, 14, 15], it assigns different fixed class labels to training samples of different classes. Then it employs the least squares regression algorithm to achieve a mapping that is able to transform training samples into approximations of their class labels. Finally CLSR uses the obtained mapping to predict the class label of every test sample. In addition to classification problems, least squares regression is also applied to subspace segmentation [16], matrix recovery [17], and feature selection [18].

The sparse representation classification (SRC) [19–21], recently proposed, can be regarded as a special form of least squares regression. Differing from LSR, it achieves an approximation of a test sample via a sparse linear combination of all training samples. Also collaboration representation [22] and

linear regression classification [23] are similar. An overview of sparse representation is provided in [24]. However, for classification tasks, because SRC must solve a set of equations for classifying every sample, CLSR is computationally much more efficient than SRC.

Xiang et al. proposed discriminative least squares regression (DLSR) [25]. The core idea is, under the conceptual framework of least squares regression, to achieve a larger class margin than the class margin obtained using CLSR for classification algorithms by using the ε dragging technique, which plays a similar role in enlarging the margin as other large margin classifiers proposed in [26–28]. The idea of using slack variable to relax the model has been widely used in the related field [29]. When the distribution of training samples is in accordance with that of test samples, the classifier learned from training samples can well adapt to test samples. Under the condition, since the classifier learned from training samples has a very large class margin, it can also obtain a satisfactory class margin for test samples. Accordingly the original ε dragging technique can perform well. In other words, a high classification accuracy can be produced. However, in real-world applications, owing to the noise or deformability of the object, the difference between training samples and test samples from the same class may be much. For example, it is well known that face images are a kind of deformable objects (owing to varying poses, expressions, and illumination conditions). Two-face images from the same subject have much difference. This difference may be even greater than that of two-face images obtained from two distinctive subjects. In this case, a large margin classifier obtained by using training samples is not usually suitable for test samples. In other words, it probably performs badly in classifying the test samples. On the contrary, reducing the class margin usually achieves better classification accuracy for classification problems on noised data. Thus, we focus on determining a proper margin by using the negative ε dragging technique and producing a robust classifier for pattern classification on noised and deformable data.

Furthermore, we focus on introducing the kernel trick to improve the ε dragging linear regression. In machine learning, the kernel trick is originally utilized to construct nonlinear support vector machines (SVMs) [30–32]. In the last more than 10 years, many kernel based approaches have been proposed, such as well-known kernel principal component analysis (KPCA) [33, 34] and kernel Fisher discriminant analysis (KFDA) [35]. For classification, Yu et al. presented the kernel nearest neighbor (KERNEL-NN) classifier [36]. KERNEL-NN applies the nearest neighbor classification method in the high dimensional feature space. The KERNEL-NN classifier could perform better than the NN classifier by utilizing an appropriate kernel. Kernel sparse representation classification (KSRC) is also presented [37, 38]. So far, by using kernel tricks [39], almost all linear learning methods can be generalized to the corresponding nonlinear ones. The kernel trick [40] goes a large step toward the goal of classifying heterogeneous data. These kernel based algorithms improve the computational ability of the linear algorithms. They first implicitly map the data in the input space into a high or even infinite dimensional kernel feature

space [18, 41] by a nonlinear mapping and then perform linear processing in the kernel feature space by using the inner products, which can be computed by a kernel function. As a result, these kernel based algorithms perform a nonlinear transformation with respect to the input space.

As is well known, kernel approach can change the distribution of samples by the nonlinear mapping. If an appropriate kernel function is utilized, kernel approach is able to make the data of different classes more linearly separable. Therefore, kernel based algorithms can perform classification well. This motivated us to integrate kernel method into linear regression for classification. If an appropriate kernel function is utilized, more samples from the same class are close to each other and samples from distinct classes are far from each other in the high dimensional feature space. Hence, in the high dimensional feature space, it is easy to learn a mapping that can well convert training samples into their class labels. Namely, linear transformation matrix learned in the high dimensional feature space can more appropriately map samples into their class labels and has more powerful discriminating ability.

Based on the above two aspects, we propose the kernel negative ε dragging linear regression (KNDLR) method in this paper. For KNDLR, samples are implicitly mapped into a high dimensional feature space first, and then linear regression with the negative ε dragging is performed in this new feature space. We prove that KNDLR in the high dimensional feature space can be formulated in terms of the inner products, while the inner products could be computed by kernel function. Thus KNDLR is easy to be implemented and has low computation cost. The classifier can generalize well because we propose and use the negative ε dragging technique, and kernel approach is also integrated into KNDLR. Comprehensive experiments demonstrated the superior characteristics of KNDLR. In summary, the contributions of the proposed method are as follows.

(1) It relaxes the strict binary label matrix that is used in conventional LR into a slack variable matrix which has more freedom to fit the sample. The proper margins between different classes are achieved by using the negative ε dragging technique. Previously researchers usually focus on enlarging the margin between different classes, whereas the negative ε dragging technique proposed by us seems to be a new contrary idea, which is useful to overcome the overfitting problem and to enhance the robustness of the algorithm on unseen samples, for example, test samples.

(2) The kernel approach is also integrated into our method. We show that KNDLR in the high dimensional feature space can be formulated in terms of the inner products, and the inner products could be computed by the kernel function. Thus KNDLR only needs to calculate the kernel function rather than directly calculating data in the high dimensional feature space corresponding to the kernel function.

(3) An algorithm named KNDLR is devised for the proposed method. The validity of the algorithms is tested on six image datasets.

The other parts of the paper are organized as follows. Section 2 briefly reviews works related to this paper. In

Section 3, our method is presented. In Section 4, analysis of our method is provided. Experimental results are reported in Section 5. Finally, Section 6 offers the conclusion of this paper.

2. Related Works

In this section, we first introduce the CLSR for classification. Then, the kernel trick is briefly reviewed.

2.1. Conventional Least Squares Regressions for Classification.

The collection of n training samples is represented as a matrix $X = [x_1, \dots, x_n]^T \in \mathbb{R}^{n \times m}$. x_i is a training sample in the form of column vector. If the training sample is a two-dimensional image, then it is converted into one column vector in advance. The objective function of conventional least squares regression (CLSR) for classification is as follows:

$$\min_W \|XW - Y\|_F^2 + \lambda \|W\|_F^2, \quad (1)$$

where $Y = [y_1, \dots, y_n]^T \in \mathbb{R}^{n \times c}$ ($c \geq 2$ is the number of class) is the binary class label matrix and the i th row y_i of Y is the class label vector of the i th sample.

For a three-class classification problem, in CLSR the class label matrix of four samples may be

$$Y = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix} \in \mathbb{R}^{4 \times 3}. \quad (2)$$

Y indicates that the first and second samples are from the first class, the third sample is from the third class, and the fourth sample is from the second class. W is the transformation matrix which converts the sample matrix X into the class label binary matrix Y . $\|\cdot\|_F^2$ stands for Frobenius norm of matrix. In the above CLSR for classification, the class label is predefined and fixed.

2.2. Kernel Trick. The kernel trick is a very powerful technique in machine learning. It has been successfully applied to many methods, such as SVM [31, 32], KPCA [33, 34], and KFDA [35]. By using kernel tricks, a linear algorithm can be easily generalized to a nonlinear algorithm.

Mercer kernel is generally used in kernel methods. It is a continuous, symmetric, positive semidefinite kernel function. Given a Mercer kernel $k : \chi \times \chi \rightarrow \mathbb{R}$, there is a unique associated reproducing kernel Hilbert space (RKHS) H_k . Usually, a Mercer kernel can be expressed as

$$k(x, x') = \Phi(x)^T \Phi(x'), \quad (3)$$

where T denotes the transpose of a matrix or vector, x and x' are any two points in χ , and Φ is the implicit nonlinear mapping associated with the kernel function $k(\cdot, \cdot)$. When implementing kernel methods, we do not need to know what Φ is and just adopt the kernel function defined as (3). Here the kernel function is the connection between the learning algorithm and data. The linear kernels, polynomial kernels,

Gaussian radial basis function (RBF) kernels, and wavelet kernels [18, 40, 41] are commonly used kernels in kernel methods. The polynomial kernel has the form of

$$k(x, x') = (x^T x' + c)^d, \quad (4)$$

where c is a constant, d is the order of polynomial, and RBF kernels can be expressed as

$$k(x, x') = \exp\left(-\gamma \|x - x'\|_2^2\right), \quad (5)$$

where γ is the parameter for RBF kernels and $\|x - x'\|_2$ is the distance between two vectors.

3. Our Method

3.1. Solving the Optimization Model. Training samples $\{x_1, x_2, \dots, x_n\}$ in the input space χ are represented as a matrix $X = [x_1, \dots, x_n]^T \in \mathbb{R}^{n \times m}$. Let Φ be the nonlinear mapping function corresponding to a kernel $k(\cdot, \cdot)$. Firstly, we implicitly employ Φ to map the data from input space χ to a high dimensional kernel feature space \mathbb{R}^f . We have

$$\Phi(X) = [\Phi(x_1), \Phi(x_2), \dots, \Phi(x_n)]^T \in \mathbb{R}^f. \quad (6)$$

Then, for classification, we should transform samples set $\Phi(X)$ to a class label matrix. But the class label matrix Y in CLSR is a strict binary label matrix which has less freedom to fit the samples. It is expected that the original strict binary constraints in Y can be relaxed into the soft constraint so that it has more freedom to fit the samples and simultaneously produce a classifier with well generalization. To this end, the slack variable matrix Y_n which is different from Y_d in DLSR is used to substitute for the original class label matrix Y . The four samples in Section 2.1 are also taken as an example here and then the slack variable class label matrix Y_n is defined as follows:

$$Y_n = \begin{bmatrix} 1 & \varepsilon_{12} & \varepsilon_{13} \\ 1 & \varepsilon_{22} & \varepsilon_{23} \\ \varepsilon_{31} & \varepsilon_{32} & 1 \\ \varepsilon_{41} & 1 & \varepsilon_{43} \end{bmatrix}, \quad (7)$$

$$\text{s.t. } \varepsilon_{ij} \geq 0.$$

It can be seen that Y_n can help to properly reduce the class margins of CLSR to generalize well. Formally, let M be a dragging matrix and defined as

$$M = \begin{bmatrix} \varepsilon_{11} & \cdots & \varepsilon_{1c} \\ \vdots & m_{ij} & \vdots \\ \varepsilon_{n1} & \cdots & \varepsilon_{nc} \end{bmatrix} \quad (8)$$

$$(i = 1, \dots, n; j = 1, \dots, c; \varepsilon_{ij} \geq 0).$$

Meanwhile, let B_n be the dragging coefficient matrix and defined as

$$B_{nij} = \begin{cases} 0 & \text{if } Y_{ij} = 1 \\ 1 & \text{if } Y_{ij} = 0; \end{cases} \quad (9)$$

then $Y_n = Y + B_n \odot M$, where \odot is a Hadamard product operator of matrices. Relaxing Y into Y_n has an idea opposite to that of the ε dragging technique in DLSR; therefore we call this relaxation the negative ε dragging.

By virtue of the kernel feature space R^f , our method tries to construct a bridge between $\Phi(X)$ and Y_n . In particular, our goal is to learn a linear function W that makes $\Phi(X)W = Y_n$ be approximately satisfied. Thus our method has the following objective function:

$$\begin{aligned} (W^*, M^*) = \arg \min_{W, M} \quad & \|\Phi(X)W - (Y + B_n \odot M)\|_F^2 \\ & + \lambda \|W\|_F^2 \\ \text{s.t.} \quad & M \geq 0, \end{aligned} \quad (10)$$

where W is the transform matrix and λ is a positive regularization parameter.

Since Y is relaxed into Y_n , (10) has more freedom than (1) to fit the samples. Based on the knowledge of Linear Algebra, we know that

$$\begin{aligned} & \|\Phi(X)W - Y_n\|_F^2 + \lambda \|W\|_F^2 \\ & = \text{trace} \left((\Phi(X)W - Y_n)(\Phi(X)W - Y_n)^T \right. \\ & \quad \left. + \lambda WW^T \right). \end{aligned} \quad (11)$$

It is easy to prove that objective function (10) is convex. Thus it has a unique solution. An iterative updating algorithm is devised to solve it. The first step of the algorithm is to solve W by fixing M .

Theorem 1. *Given M , the optimal W in (10) can be calculated as*

$$W^* = (\Phi(X)^T \Phi(X) + \lambda I)^{-1} \Phi(X)^T Y_n. \quad (12)$$

Proof. According to matrix theory, the optimal W can be obtained by making the derivation of (10) with respect to W and set it to zero. That is,

$$\begin{aligned} & \frac{\partial (\|\Phi(X)W - Y_n\|_F^2 + \lambda \|W\|_F^2)}{\partial W} \\ & = \Phi(X)^T \Phi(X)W - \Phi(X)^T Y_n + \lambda W = 0 \implies \end{aligned} \quad (13)$$

$$W^* = (\Phi(X)^T \Phi(X) + \lambda I)^{-1} \Phi(X)^T Y_n.$$

The second step of our algorithm is to solve M by fixing W . Then (10) can be rewritten as $\arg \min_M \|\Phi(X)W - (Y + B_n \odot M)\|_F^2$. M can be obtained by solving the following optimization problem:

$$\begin{aligned} \min_M \quad & \|G - B_n \odot M\|_F^2 \\ \text{s.t.} \quad & M \geq 0, \end{aligned} \quad (14)$$

where

$$G = \Phi(X)W^* - Y. \quad (15)$$

Considering the i th row and j th column element M_{ij} of M , we have

$$\begin{aligned} \min_{M_{ij}} \quad & (G_{ij} - B_{nij}M_{ij})^2 \\ \text{s.t.} \quad & M_{ij} \geq 0. \end{aligned} \quad (16)$$

According to [25], the formula to calculate M_{ij} is

$$M_{ij} = \max(B_{nij}G_{ij}, 0). \quad (17)$$

Therefore, the optimal solution of M is

$$M = \max(B_n \odot G, 0). \quad (18)$$

In a word, the first step of the algorithm is to solve W by fixing M , and the second step of the algorithm is to solve M by fixing W . In other words, (12) should be calculated in the first step, and (15) and (18) should be calculated in the second step. These two steps should be repeatedly calculated till the termination condition is satisfied. \square

3.2. Integrating the Kernel Trick into the Optimization Model.

As mentioned above, we should repeatedly calculate (12) and (18). However, for (12) and (18), $\Phi(X)$ exists in kernel feature space R^f . Fortunately, we do not need to know what $\Phi(X)$ is and just adopt the kernel function (3). How to use the kernel function to eliminate denotation $\Phi(X)$ is presented as follows.

Let

$$P = (\Phi(X)^T \Phi(X) + \lambda I)^{-1} \Phi(X)^T. \quad (19)$$

By using the following formula [42] on matrix manipulations:

$$(A^{-1} + B^T R^{-1} B)^{-1} B^T R^{-1} = AB^T (BAB^T + R)^{-1}, \quad (20)$$

we use λI , $\Phi(X)$, and I instead of A^{-1} , B , and R , respectively, having

$$\begin{aligned} & (\lambda I + \Phi(X)^T \Phi(X))^{-1} \Phi(X)^T \\ & = \Phi(X)^T (\Phi(X) \Phi(X)^T + \lambda I)^{-1}. \end{aligned} \quad (21)$$

Then, we substitute it into (12); therefore

$$W^* = \Phi(X)^T (\Phi(X) \Phi(X)^T + \lambda I)^{-1} Y_n \implies \quad (22)$$

$$W^* = \Phi(X)^T (K + \lambda I)^{-1} Y_n = PY_n, \quad (23)$$

where $K_{i,j} = \Phi(x_i)^T \Phi(x_j) = k(x_i, x_j)$, ($i, j = 1, 2, \dots, n$).

Actually, $\Phi(X)^T (K + \lambda I)^{-1}$ in (23) is changeless because it only depends on X and the utilized kernel function, while Y_n is changeable during the iteration; hence, for avoiding to directly calculate $\Phi(X)$, in the first step we only need to calculate

$$Y_n = Y + B_n \odot M. \quad (24)$$

Input: Training samples matrix X ; Label matrix Y ; dragging coefficient matrix B_n ; test sample t ; parameter λ ;

Output: the slack variable class label matrix Y_n ; predicted class k for test sample t ;

Initialization: $M = 0_{n \times c}$;
Calculate $H = K(K + \lambda I)^{-1}$;
Set threshold h ; Set $itr = 1$.

Repeat

- (1) Given M , calculate $Y_n = Y + B_n \odot M$.
- (2) Utilize $G = HY_n - Y$, then calculate $M = \max(B \odot G, 0)$.

Until the absolute value of the difference between objective functions of two consecutive loops is smaller than threshold h .

- (3) For test sample t , calculate $t_y = t^T W^*$.
- (4) If $k = \arg_j \max t_y^j$, then t is classified into the k th class. t_y^j is the j th entry of t_y .

Output: the transformation matrix W^* , k .

ALGORITHM 1

The second step of algorithm is to solve M by calculating (15) and (18). By substituting (23) into (15), we have

$$G = K(K + \lambda I)^{-1} Y_n - Y. \quad (25)$$

Hence, in the second step we need to calculate (25) and (18).

Then the predicted label for a test sample x is

$$Y(x) = \Phi(x)^T W. \quad (26)$$

Intuitively, W should be calculated by iteration and then it is utilized to calculate the predicted label $Y(x)$ for test sample x . However, by substituting (23) into (26), we have

$$\begin{aligned} Y(x) &= \Phi(x)^T \Phi(X)^T (K + \lambda I)^{-1} Y_n \\ &= \kappa(x) (K + \lambda I)^{-1} Y_n, \end{aligned} \quad (27)$$

where $\kappa(x) = [k(x, x_1), k(x, x_2), \dots, k(x, x_n)]$.

Because $\Phi(X)^T (K + \lambda I)^{-1}$ depends on X and the utilized kernel function, we only need to calculate Y_n out by the iteration, and after the iteration is performed, the predicted label for a test sample x can be obtained by (27). As presented above, directly calculating $\Phi(X)$ can be avoided by utilizing the kernel function.

In summary, we do not need to know what Φ is and just adopt the kernel function during the iteration. The complete algorithm is summarized in Algorithm 1.

4. Analysis of Our Method

In our method, the negative ε dragging technique and kernel trick are simultaneously integrated into the LR model to obtain more robust classification result for noised and deformable data. We analyze our method from two aspects.

Firstly, we present the class margins of our method, DLSR, and CLSR for classification. For simplicity of description, the

four samples in Section 2.1 are also taken as an example here. For our method, it is clear that

$$Y_n = \begin{bmatrix} 1 & \varepsilon_{12} & \varepsilon_{13} \\ 1 & \varepsilon_{22} & \varepsilon_{23} \\ \varepsilon_{31} & \varepsilon_{32} & 1 \\ \varepsilon_{41} & 1 & \varepsilon_{43} \end{bmatrix}. \quad (28)$$

For DLSR,

$$Y_d = \begin{bmatrix} 1 + \varepsilon_{11} & -\varepsilon_{12} & -\varepsilon_{13} \\ 1 + \varepsilon_{21} & -\varepsilon_{22} & -\varepsilon_{23} \\ -\varepsilon_{31} & -\varepsilon_{32} & 1 + \varepsilon_{33} \\ -\varepsilon_{41} & 1 + \varepsilon_{42} & -\varepsilon_{43} \end{bmatrix}. \quad (29)$$

Suppose that Y_n and Y_d have the same ε components. For the first and third samples (they, respectively, belong to the first and third classes), the distance between their class labels can be denoted by

$$d_n = \sqrt{(1 - \varepsilon_{31})^2 + (\varepsilon_{12} - \varepsilon_{32})^2 + (\varepsilon_{13} - 1)^2}. \quad (30)$$

For DLSR, the distance between the class labels of the first and third samples can be denoted by

$$\begin{aligned} d_d &= \sqrt{(1 + \varepsilon_{11} + \varepsilon_{31})^2 + (\varepsilon_{12} - \varepsilon_{32})^2 + (\varepsilon_{13} + \varepsilon_{33} + 1)^2}. \end{aligned} \quad (31)$$

For CLSR, the distance between their class labels can be denoted by $d = \sqrt{2}$.

We see that if Y_n and Y_d have same ε components, DLSR has the largest class margin whereas our method usually has the smallest class margin. In other words, we usually have $d_d \geq d \geq d_n$. Actually, because $\varepsilon_{ij} \geq 0$, it is absolutely certain

that $d_d \geq d$. As for $d \geq d_n$, it can be demonstrated below. First, $d_n = \sqrt{1 - 2\varepsilon_{31} + \varepsilon_{31}^2 + \varepsilon_{12}^2 - 2\varepsilon_{12}\varepsilon_{32} + \varepsilon_{32}^2 + 1 - 2\varepsilon_{13} + \varepsilon_{13}^2}$. Because $\varepsilon_{ij} \ll 1$ is usually satisfied, we can ignore the second-order terms and have $d_n \approx \sqrt{1 - 2\varepsilon_{31} + 1 - 2\varepsilon_{13}} \leq \sqrt{2}$. As a result, in the scenario of noised and deformable data, our method can effectively decrease the probability that the classifier learned from training samples too fits training samples and cannot be well applicable for test samples. In other words, our method can make the obtained classifier generalize well and is very suitable for the classification of noised and deformable data.

Secondly, we present effects of the kernel trick integrated into our method. In some real-world applications, samples from different classes are mixed up and are not linearly separable, because the difference between training samples from the same class may be much more than the difference between training samples from different classes. For instance, in the face recognition problem, the face images from the same person may be more different than the face images from distinct persons owing to variable expressions, poses, and illuminations. This is known as the problem of uncertain data [42, 43]. Under this situation, both CLSR and DLSR could not attain a good classification performance. The kernel approach can change the distribution of samples by the nonlinear mapping. If an appropriate kernel function is utilized, the kernel approach can make linearly nonseparable samples become linearly separable. The term linearly separable means that samples of different classes have good separability. Exactly, it is referred to as a linear boundary such as a line or plane that can separate samples from different classes without errors.

Here, kernel mapping is integrated into our approach so that, in the high dimensional kernel feature space, it is easy to learn a mapping that can well convert training samples into their class labels. Namely, the linear transformation matrix obtained in the high dimensional feature space can more appropriately map training samples into their class labels. Therefore, our kernel based approach can perform classification well.

If the two class samples are not linearly separable, CLSR and DLSR could not attain a good classification performance. KNDLR firstly makes a nonlinear mapping of the data to enhance the linear separability of samples; hence KNDLR is able to obtain higher classification accuracy than CLSR and DLSR. Moreover, our KNDLR just utilizes the kernel function to calculate transform matrix W and class label for test samples instead of directly calculating $\Phi(X)$.

In addition, the overall complexity of KNDLR is low, although it is solved iteratively. In each iteration, the main computation cost is in (25), where we need to calculate the matrix inverse $(K + \lambda I)^{-1}$. Since $K(K + \lambda I)^{-1}$ is only dependent on X and the utilized kernel function, it can be precalculated before the loop is carried out. Thus the speed of calculating G in (25) is very fast. Moreover, it is obvious that K is an $n \times n$ matrix (n is the number of training samples), while $(X^T X + \lambda I)^{-1}$ calculated in CLSR or DLSR is an $m \times m$ matrix (m is the number of features). Thus, when the number of samples is much less than the dimension of features, the size of K is small. Hence it is easy

to calculate the matrix inverse $(K + \lambda I)^{-1}$. If the features are very high dimensional, calculation of the matrix inverse $(X^T X + \lambda I)^{-1}$ will be quite time-consuming and memory-consuming. In particular, although our KNDLR approach is similar to CLSR and DLSR in some aspects, it is much more efficient than them when classifying high dimensional data. However, when the number of samples is not much smaller than that of features and dimension of features is high, size of K is large. Hence calculating the matrix inverse $(K + \lambda I)^{-1}$ is complex as solving the inverse matrix $(X^T X + \lambda I)^{-1}$ and the efficient of our KNDLR is almost the same as that of CLSR and DLSR.

5. Experiments

In our experiments, KNDLR was compared with CLSR, DLSR, NDLR (the KNDLR without kernel trick), kernel support vector machine (K -SVM) in [31], k nearest neighbor method (KNN), the nonnegative least squares method (NNLS) proposed in [3], sparse representation based classification (SRC (l1_ls)), and linear regression based classification (LRC). We use five face image databases and a handwriting digit dataset, namely, Georgia Tech (GT), FERET, LFW, AR, YaleB, and MNIST dataset. The subsets of the last two datasets, which are available at “<http://www.cad.zju.edu.cn/home/dengcai/Data/data.html>,” were used to perform our experiments. All methods were directly performed on image, with no extracting feature from image in advance. Our method, CLSR, DLSR, and NDLR all have a parameter λ . The parameter was set to 0.0001, 0.0005, 0.01, 0.02, 0.03, 0.04, 0.05, 0.06, 0.07, 0.08, 0.09, 0.1, 0.2, 0.3, 0.4, and 0.5, respectively. The best accuracy of each method is given for comparison. Threshold h is set to 0.0001. For KNDLR, we used polynomial kernel $k(x, x') = (x^T x' + c)^d$ on LFW, AR, YaleB, and MNIST and Gaussian radial basis function (RBF) kernel $k(x, x') = \exp(-\gamma \|x - x'\|_2^2)$ on GT and FERET, respectively. The parameters of polynomial kernels c and d were set to 1 and 2, respectively. The parameter of RBF kernel γ was set to the median value of $1/(\|x_i - \bar{x}\|^2)$, $i = 1, \dots, n$, where \bar{x} is the mean of all training samples. For K -SVM, package libsvm-mat-3.0-1 is used. The libsvm_options of the function “svmtrain” were set to [“-s 0 -t 2 -g 1.0e - 1”], where “-t 2” indicates Gaussian radial basis function (RBF) kernel. The value of hyperparameter C was selected from the candidate set {0.01, 0.1, 1.0, 10.0, 100.0, 1000.0} by cross-validation approach. For KNN, k was set to 1, and Euclidean distance metric was used to find the nearest neighbor.

5.1. Experiment on the GT Database. The Georgia Tech (GT) face database contains 750 images from 50 subjects. For each subject 15 face images are available. The pictures show frontal and/or tilted faces with different facial expressions, lighting conditions, and scales. Figure 1 presents some face images from the GT face database. In our experiments, all images in the database were manually cropped and resized to 30×40 . After the image cropping, most of the complex background has been excluded. They are further converted to gray level images for both training and testing purposes.



FIGURE 1: Some face images from the GT face database.

TABLE 1: Accuracies (%) of different methods on the GT database.

Number of training samples per class	5	6	7	8	9	10
Our method	70.80	73.53	76.05	80.26	81.33	82.36
DLSR	50.48	50.93	52.80	53.77	53.77	54.56
CLSR	63.58	65.33	67.30	70.57	71.00	71.40
NNLS	68.90	71.02	73.88	76.83	78.27	79.68
K-SVM	70.42	72.20	74.93	79.46	80.17	81.92
KNN	59.30	61.76	63.85	67.49	69.23	69.40
SRC	56.18	58.44	60.13	63.09	63.57	64.36
LRC	68.90	71.22	73.78	77.71	79.20	80.68
NDLR	65.66	66.76	69.38	73.60	73.57	74.56

In our experiments, we randomly took n ($= 5, 6, 7, 8, 9, 10$) face images of each subject as original training samples, respectively, and treated the remaining face images as testing samples. For each given n , we take the average value of classification rates calculated from 10 random splits as final classification rate. The experimental results are presented in Table 1. From this table, we can conclude that the proposed method obtains the best classification accuracy.

5.2. Experiment on the FERET Face Dataset. A subset of the FERET face dataset was used in the experiment. This subset includes 1442 face images from 206 subjects and each subject has seven different face images. This subset was composed of the images in the original FERET face dataset whose names are marked with two-character strings: “ba,” “bj,” “bk,” “be,” “bf,” “bd,” and “bg”. Figure 2 shows some image examples. We resized all face images to 40 by 40 matrices.

In our experiments, n ($= 1, 2, 3, 4, 5$) samples of each subject were randomly taken as training samples and the remaining samples were treated as test samples. For each given n , we take the average value of classification rates calculated from 10 random splits as final classification rate. Experimental results of classification accuracies are shown in Table 2. Table 2 demonstrates that our method performs better than the other methods.

TABLE 2: Accuracies (%) of different methods on the FERET database.

Number of training samples per class	1	2	3	4	5
Our method	35.52	55.84	66.29	79.03	78.18
DLSR	19.45	31.18	36.99	45.95	44.15
CLSR	28.03	46.38	55.18	65.03	65.07
NNLS	34.26	53.48	64.38	75.84	76.33
K-SVM	32.01	54.73	64.43	75.89	73.96
KNN	32.01	44.87	54.44	68.14	65.70
SRC	23.54	41.48	53.12	68.51	70.46
LRC	32.01	55.34	65.89	77.12	74.76
NDLR	33.77	54.28	63.50	72.64	72.60

5.3. Experiment on the LFW Face Dataset. The LFW dataset is a face image dataset for unconstrained face recognition. Images in this dataset vary much in clothing, pose, and background more than the other face datasets. There are more than 13000 faces images collected from the web. Every face image is manually labeled. We use only a subset composed of 1251 images from 86 subjects to conduct experiments. Figure 3 shows some example face images. Each image is cropped and resized to 32×32 image.

A random subset with n ($= 6, 7, 8, 9, 10$) images per individual was taken with labels to form the training set, and the rest of the database was considered to be the testing set. For each given n , there are 10 random splits. The average value of classification rates calculated from 10 random splits was taken as final classification rate. The classification accuracies were shown in Table 3. It is clear that our method performs better than the rest of methods.

5.4. Experiment on the AR Face Dataset. The AR dataset contains over 4000 face images of 126 subjects, including frontal views of faces with different facial expressions, lighting conditions, and occlusion. We use only a subset composed of 3120 images from 120 subjects and each subject has 26 different face images. Figure 4 shows some example face images. Each image is cropped and resized to 40×50 image.



FIGURE 2: Some face images from the FERET face database.



FIGURE 3: Some face images from the LFW face database.

A random subset with n ($= 1, 2, 3, 4, 5$) images per individual was taken with labels to form the training set, and the rest of the database was considered to be the testing set. For each given n , there are 10 random splits. The average classification rate calculated from 10 random splits was taken as final classification rate. The classification accuracies were shown in Table 4. It is clear that our method performs better than the rest of methods.

5.5. Experiment on the YaleB Face Dataset. For this database, we simply use the cropped images and resize them to 32×32 pixels to conduct experiments. Figure 5 shows some example face images.

A random subset with n ($= 5, 6, 7, 8$) images per individual was taken with labels to form the training set, and the rest of the database was considered to be the testing set. For each given n , there are 10 random splits. The average



FIGURE 4: Some face images from the AR face database.

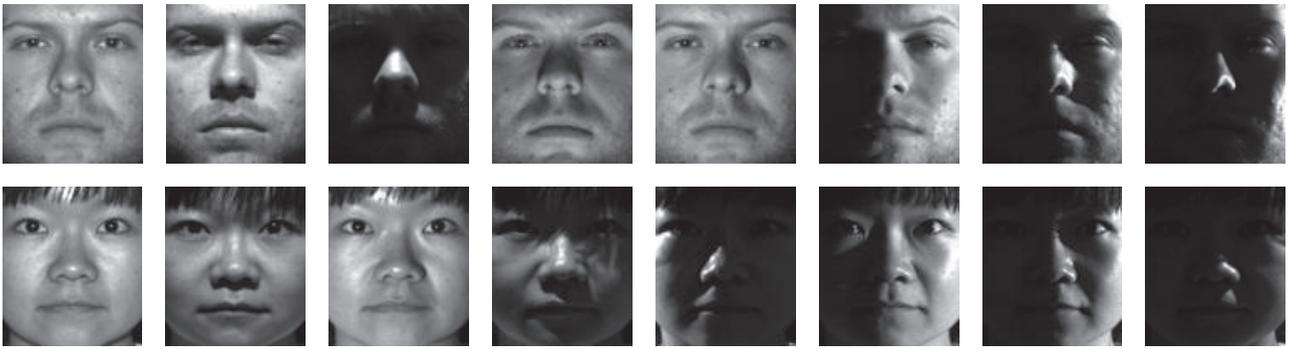


FIGURE 5: Some face images from the YaleB face database.

TABLE 3: Accuracies (%) of different methods on the LFW database.

Number of training samples per class	6	7	8	9	10
Our method	35.95	38.00	39.13	40.27	41.10
DLSR	14.46	14.92	14.58	15.43	14.68
CLSR	34.45	35.93	35.15	36.65	35.27
NNLS	33.86	36.58	36.04	36.69	37.31
K-SVM	32.88	34.76	35.70	36.52	38.59
KNN	20.34	21.28	21.12	21.26	21.53
SRC	34.15	36.70	38.06	39.31	39.00
LRC	32.12	35.16	34.78	36.50	40.05
NDLR	35.22	37.49	38.40	39.29	40.69

classification rate calculated from 10 random splits was taken as final classification rate. The classification accuracies were shown in Table 5. It is clear that our method performs better than the rest of methods, except for SRC. However, SRC is time-consuming, which is shown in Section 5.7.

5.6. Experiment on the MNIST Dataset. The MNIST database of handwritten digits from Yann LeCun's page has a training set of 60,000 examples and a test set of 10,000 examples. We use only a subset composed of the first 2k training images and first 2k test images to conduct experiments. The size of each image is 28×28 pixels, with 256 gray levels per pixel.

TABLE 4: Accuracies (%) of different methods on the AR database.

Number of training samples per class	1	2	3	4	5
Our method	64.69	75.87	83.74	90.46	95.12
DLSR	56.83	70.01	78.34	85.89	91.37
CLSR	59.73	74.00	82.55	89.05	93.89
NNLS	59.93	67.08	78.03	85.09	92.90
K-SVM	58.95	66.39	74.56	83.64	91.69
KNN	58.95	61.45	68.88	75.37	84.03
SRC	61.81	74.36	82.72	89.95	94.35
LRC	58.95	65.42	73.94	82.06	91.15
NDLR	64.00	74.87	83.38	90.02	94.63

Thus, each image is represented by a 784-dimensional vector. Figure 6 shows some example images. Experimental results of classification accuracies are shown in Table 6. From this table, we can conclude that the proposed method obtains the best classification accuracy.

5.7. Computing Time. Aforementioned experiments were performed on an Intel machine (Core (TM) i5-6600 CPU, 3.30 GHz, 8 GB RAM, with 64-bit Win 10 Chinese operating system). All methods, except for the SVM methods, were implemented by software MATLAB 2010a. The libSVM3.0 toolbox in the language C was utilized for performing SVM.

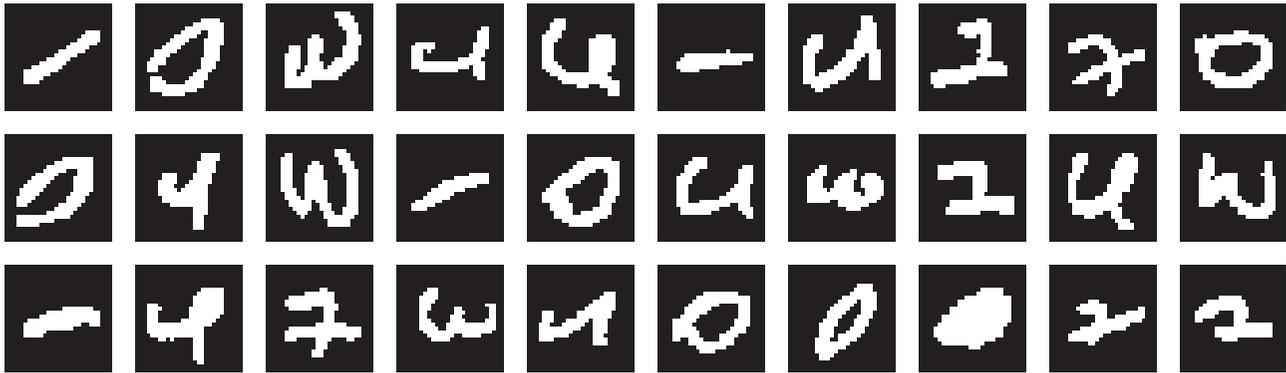


FIGURE 6: Some images from the MNIST database.

TABLE 5: Accuracies (%) of different methods on the YaleB database.

Number of training samples per class	5	6	7	8
Our method	80.48	81.67	82.28	84.55
DLSR	76.05	78.48	79.33	77.54
CLSR	76.83	78.87	80.11	77.85
NNLS	68.98	73.39	73.38	75.14
K -SVM	74.45	76.53	75.58	76.93
KNN	57.56	60.42	60.14	60.92
SRC	82.53	84.54	85.22	85.29
LRC	79.14	81.01	80.87	84.36
NDLR	77.02	79.40	80.25	78.73

Besides classification accuracies, because the computing time is significantly different for each method, we select the experiment on GT and AR to show the computing time of each method. The GT database only contains a small number of samples, while the AR database contains a relatively large number of samples, which represent two different cases. Here, the computing time of each method is the sum of time spent on learning from samples and time spent on classification of new samples when training samples and test samples have been given. We use MATLAB instruction tic and toc to get the time. Table 7 shows the computing time of the methods on GT and Table 8 shows that on AR.

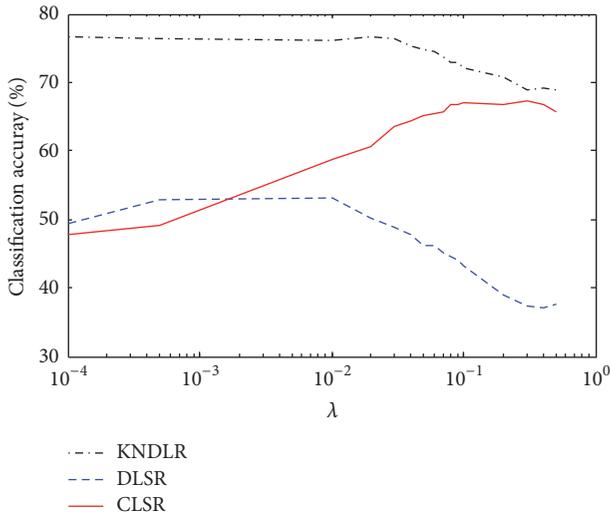
First, it can be clearly seen that our KNDLR approaches are very fast as DLSR, CLSR, NDLR, SVM, and KNN on GT having a small number of samples and AR having relatively large number of samples. Second, KNDLR is much faster than NNLS and SRC, especially on AR. Third, it is shown that the computing time of KNDLR on AR is only a little longer than that on GT, while the computing time of some methods on AR, such as NNLS and SRC, is far longer than that on GT. In particular, SRC becomes very time-consuming when the number of samples is large. One of the reasons for the efficiency of our methods is that, in our approach, the procedure of learning is executed only once, then the results are saved for classifying all new samples. SRC needs

to learn a linear combination of all training samples for every new sample; thus when the number of samples is large, SRC becomes extremely time-consuming. This demonstrates that our KNDLR is efficient.

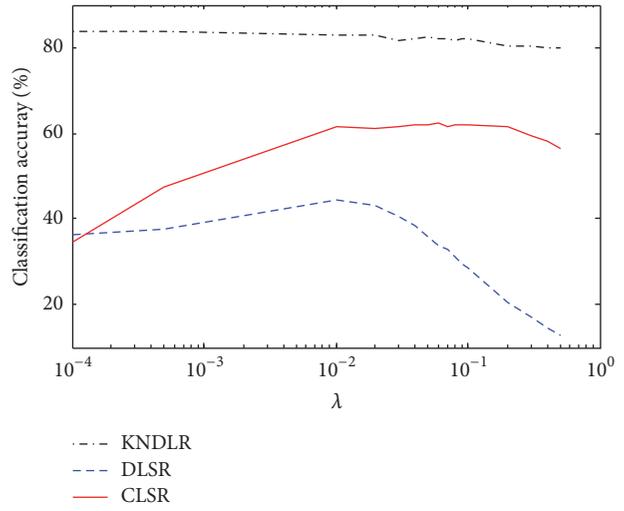
5.8. Parameter λ and Convergence. In order to further illustrate the properties of KNDLR, the classification accuracies corresponding to different values of λ and convergence are shown in Figures 7 and 8, respectively, where ($\#n$) represents that the first n samples were utilized for training and the remaining for testing. KNDLR, DLSR, and CLSR are similar to each other to some degree. All of them apply the least squares regression and have a regularization parameter λ . In Figure 7, it is shown that KNDLR is relatively more robust to λ than DLSR and CLSR. Especially, for GT, FERET, AR, and MNIST, the classification accuracies obtained by utilizing KNDLR vary in a small range. It is also observed that a relative large value of λ cannot bring more preferable classification accuracy and λ could be limited to $[10^{-4}, 0.5]$. In real application, the cross-validation method is utilized to determine the optimal value of λ from this range. More importantly, in Figure 8, it is shown that KNDLR converges very fast on six datasets, especially on FERET database.

6. Conclusions

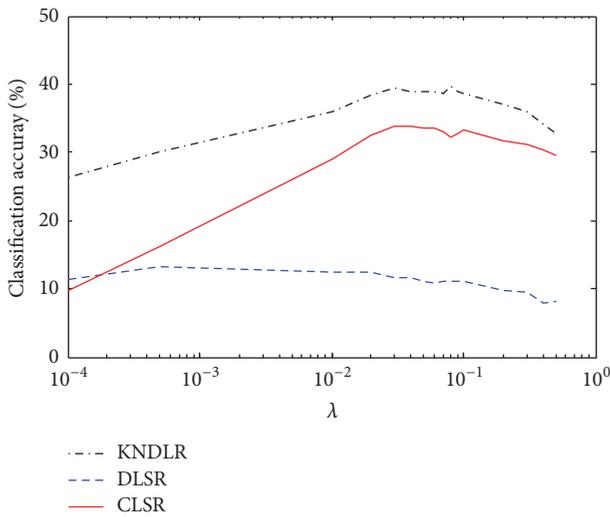
This paper proposed a kernel negative ε dragging linear regression method for pattern classification, which simultaneously integrated the negative ε dragging technique and the kernel method into linear regression for robust pattern classification under the condition that the consistency and compatibility between the test samples and training samples are poor. The negative ε dragging technique learns a classifier with a proper margin from noised and deformable data. Meanwhile, the kernel approach can make linearly nonseparable samples become linearly separable. Based on effect of the negative ε dragging technique and kernel collaboration, our method can better perform classification for noised and deformable data. Comprehensive experiments on six different datasets demonstrate that proposed KNDLR outperforms existing LR method for classification and some other commonly used methods such as SVM, NNLS, SRC, and LRC, and our KNDLR is efficient.



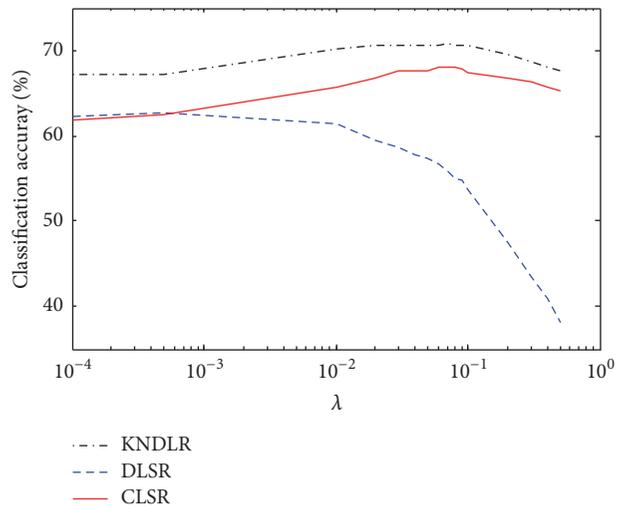
(a) GT (#8)



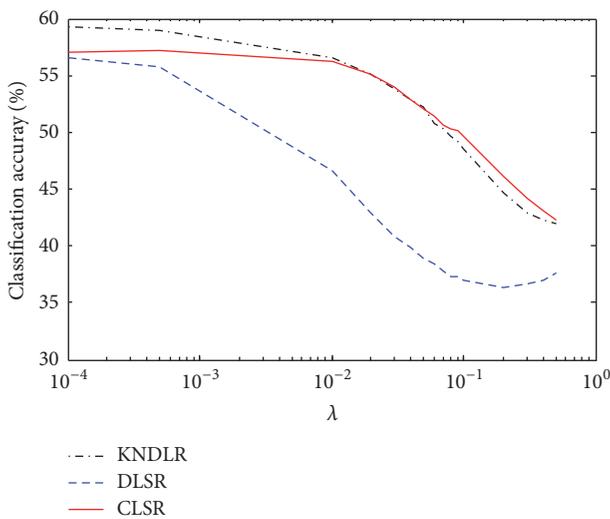
(b) FERET (#5)



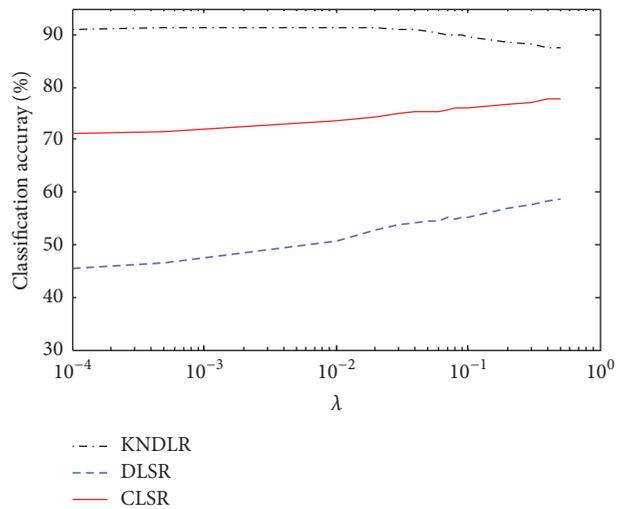
(c) LFW (#8)



(d) AR (#3)



(e) YaleB (#7)

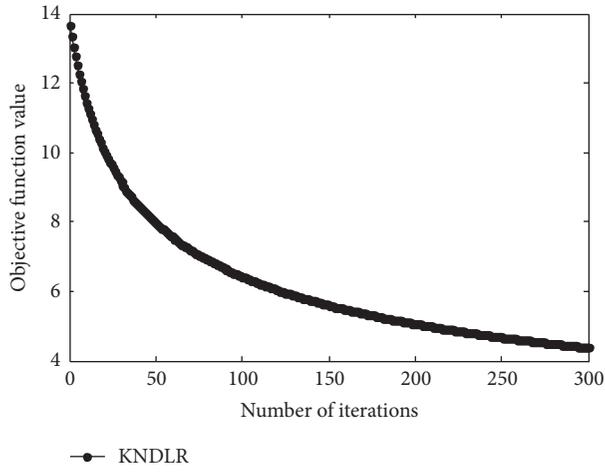


(f) MNIST

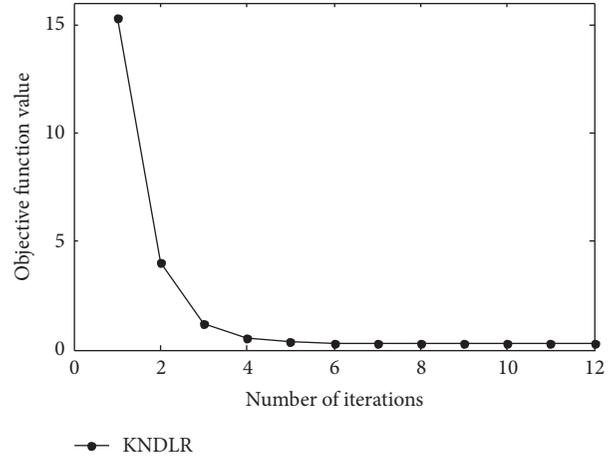
FIGURE 7: The classification accuracies (%) versus value λ on the six datasets.

TABLE 6: Accuracies (%) of different methods on the MNIST database.

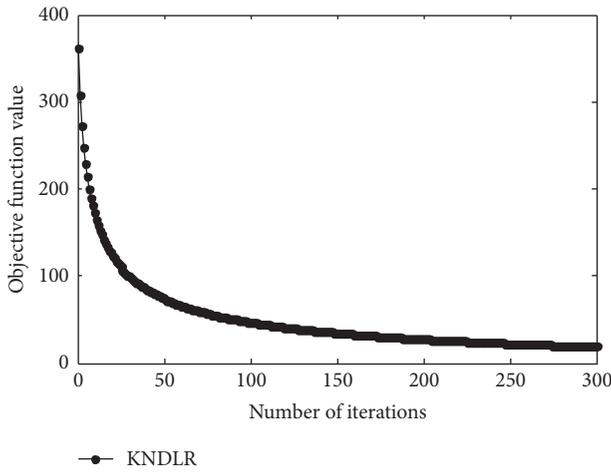
Alg.	Our method	DLSR	CLSR	NNLS	K-SVM	KNN	SRC	LRC	NDLR
Accur.	91.5	58.60	77.70	90.25	86.25	88.90	84.50	82.70	79.00



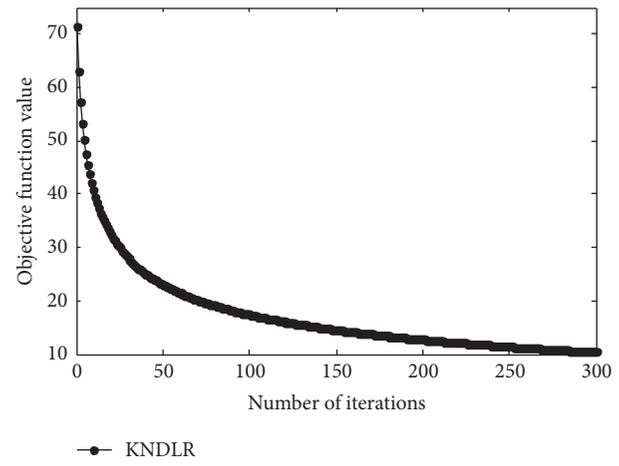
(a) GT (#8)



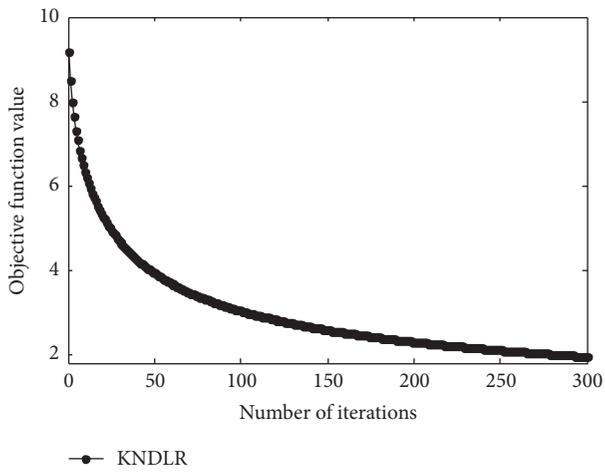
(b) FERET (#5)



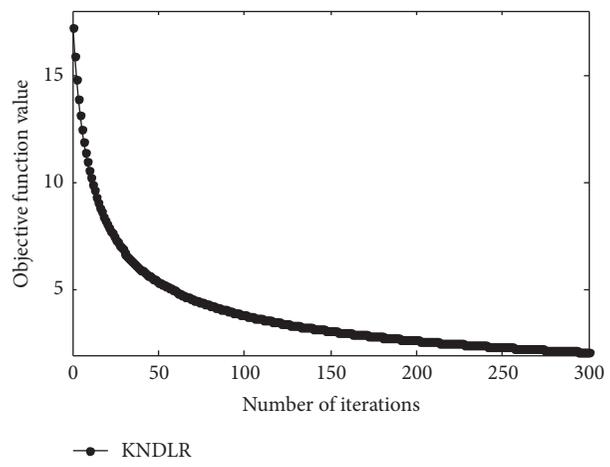
(c) LFW (#8)



(d) AR (#3)



(e) YaleB (#7)



(f) MNIST

FIGURE 8: Convergence curves of KNDLR on the six datasets.

TABLE 7: Computing time (s) of different methods on the GT database.

Number of training samples per class	5	6	7	8	9	10
Our method	1.94	2.41	2.72	3.24	3.69	4.21
DLSR	3.67	4.01	4.06	4.24	4.46	4.65
CLSR	1.80	2.01	1.80	1.85	1.84	1.88
NNLS	7.56	10.71	13.54	16.46	18.99	22.08
SVM	0.41	0.50	0.60	0.70	0.80	0.90
KNN	0.75	0.80	0.85	0.85	0.88	0.80
SRC	63.94	77.34	94.32	103.43	115.84	124.28
LRC	3.63	3.44	3.58	3.44	3.52	3.35
NDLR	3.72	3.90	4.06	4.25	4.39	4.60

TABLE 8: Computing time (s) of different methods on the AR database.

Number of training samples per class	1	2	3	4	5
Our method	2.88	5.76	8.62	12.10	14.74
DLSR	18.28	19.69	21.17	22.76	24.20
CLSR	10.30	10.46	10.62	10.73	10.84
NNLS	14.70	50.14	120.81	251.74	447.02
SVM	1.41	2.55	3.67	4.88	6.04
KNN	3.16	6.24	10.18	23.05	23.99
SRC	231.48	803.65	1564.73	2375.6	3616.28
LRC	21.83	32.54	41.38	48.06	62.64
NDLR	18.25	19.81	21.16	22.73	24.19

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (nos. 61672333, 61402274, and 41471280), the Program of Key Science and Technology Innovation Team in Shaanxi Province (no. 2014KTC-18), the Key Science and Technology Program of Shaanxi Province, China (no. 2016GY-081), the Fundamental Research Funds for the Central Universities (no. 2017CSY024), the Industry University Cooperative Education Project of Higher Education Department of the Ministry of Education (no. 201701023062), and the Interdisciplinary Incubation Project of Learning Science of Shaanxi Normal University.

References

- [1] T. Strutz, *Data fitting and uncertainty: a practical introduction to weighted least squares and beyond*, Vieweg, Wiesbaden, Germany, 2010.
- [2] S. Wold, A. Ruhe, H. Wold, and W. J. Dunn III, "The collinearity problem in linear regression. The Partial Least Squares (PLS) approach to generalized inverses," *SIAM Journal on Scientific and Statistical Computing*, vol. 5, no. 3, pp. 735–743, 1984.
- [3] Y. Li and A. Ngom, "Nonnegative least-squares methods for the classification of high-dimensional biological data," *IEEE Transactions on Computational Biology and Bioinformatics*, vol. 10, no. 2, pp. 447–456, 2013.
- [4] Y. Xu, X. Fang, Q. Zhu, Y. Chen, J. You, and H. Liu, "Modified minimum squared error algorithm for robust classification and face recognition experiments," *Neurocomputing*, vol. 135, pp. 253–261, 2014.
- [5] C. Zhou, W. Lu, P. Zhang, J. Wu, Y. Hu, and L. Guo, "On the minimum differentially resolving set problem for diffusion source inference in networks," in *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, pp. 79–85, 2016.
- [6] C. Xu, D. Tao, Y. Li, and C. Xu, "Large-margin multi-view Gaussian process," *Multimedia Systems*, vol. 21, no. 2, pp. 147–157, 2014.
- [7] U. Brefeld, T. Gärtner, T. Scheffer, and S. Wrobel, "Efficient co-regularized least squares regression," in *Proceedings of the Twenty-Third International Conference (ICML 2006)*, pp. 137–144, Carnegie Mellon University, Pittsburgh, USA, June 2006.
- [8] B. Du, M. Zhang, L. Zhang, R. Hu, and D. Tao, "PLTD: Patch-Based Low-Rank Tensor Decomposition for Hyperspectral Images," *IEEE Transactions on Multimedia*, vol. 19, no. 1, pp. 67–79, 2017.
- [9] T. Liu, M. Gong, and D. Tao, "Large-Cone Nonnegative Matrix Factorization," *IEEE Transactions on Neural Networks and Learning Systems*, 2016.
- [10] C. Liu, C. Zhou, J. Wu, H. Xie, Y. Hu, and L. Guo, "CPMF: A collective pairwise matrix factorization model for upcoming event recommendation," in *Proceedings of the 2017 International Joint Conference on Neural Networks (IJCNN)*, pp. 1532–1539, Anchorage, AK, USA, May 2017.
- [11] S. M. Xiang, F. P. Nie, and C. S. Zhang, "Regression reformulations of LLE and LTSA with locally linear transformation," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 41, no. 5, pp. 1250–1262, 2011.
- [12] F. P. Nie, H. Wang, H. Huang, and C. Ding, "Adaptive loss minimization for semi-supervised elastic embedding," in *Proceedings of the in proceedings of the twenty-third international joint conference on artificial intelligence*, 2013.
- [13] X. Z. Fang, Y. Xu, X. Li, Z. Lai, and W. K. Wong, "Learning a non-negative sparse graph for linear regression," *IEEE Transactions on Image Processing*, vol. 24, no. 9, pp. 2760–2771, 2015.
- [14] S. Liu, L. Li, Y. Peng, G. Qiu, and T. Lei, "Improved sparse representation method for image classification," *IET Computer Vision*, vol. 11, no. 4, pp. 319–330, 2017.
- [15] T. Strutz, *Data Fitting and Uncertainty*, Vieweg, Wiesbaden, Germany, 2010.
- [16] B. Du and L. Zhang, "A discriminative metric learning based anomaly detection method," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 52, no. 11, pp. 6844–6857, 2014.
- [17] T. Liu and D. Tao, "On the performance of Manhattan nonnegative matrix factorization," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 27, no. 9, pp. 1851–1863, 2016.
- [18] B. Du, W. Xiong, J. Wu, L. Zhang, L. Zhang, and D. Tao, "Stacked Convolutional Denoising Auto-Encoders for Feature Representation," *IEEE Transactions on Cybernetics*, vol. 47, no. 4, pp. 1017–1027, 2017.
- [19] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, and Y. Ma, "Robust face recognition via sparse representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 2, pp. 210–227, 2009.

- [20] Y. Xu, D. Zhang, J. Yang, and J. Y. Yang, "A two-phase test sample sparse representation method for use with face recognition," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 21, no. 9, pp. 1255–1262, 2011.
- [21] Y. Xu, Q. Zhu, Z. Fan, D. Zhang, J. Mi, and Z. Lai, "Using the idea of the sparse representation to perform coarse-to-fine face recognition," *Information Sciences*, vol. 238, pp. 138–148, 2013.
- [22] S. Liu, Y. Peng, X. Ben, W. Yang, and G. Qiu, "A novel label learning algorithm for face recognition," *Signal Processing*, vol. 124, pp. 141–146, 2016.
- [23] I. Naseem, R. Togneri, and M. Bennamoun, "Linear regression for face recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 11, pp. 2106–2112, 2010.
- [24] Z. Zhang, Y. Xu, J. Yang, X. Li, and D. Zhang, "A survey of sparse representation: algorithms and applications," *IEEE Access*, vol. 3, pp. 490–530, 2015.
- [25] S. Xiang, F. Nie, G. Meng, C. Pan, and C. Zhang, "Discriminative least squares regression for multiclass classification and feature selection," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 23, no. 11, pp. 1738–1754, 2012.
- [26] Q. Li, B. Xie, J. You, W. Bian, and D. Tao, "Correlated logistic model with elastic net regularized for multilabel image classification," *IEEE Transactions on Image Processing*, vol. 25, no. 8, pp. 3801–3813, 2016.
- [27] C. Gong, D. Tao, W. Liu, L. Liu, and J. Yang, "Label propagation via teaching-to-learn and learning-to-teach," *IEEE Transactions on Neural Networks Learning Systems*, vol. 28, no. 6, pp. 1452–1465, 2017.
- [28] C. Gong, T. Liu, D. Tao, K. Fu, E. Tu, and J. Yang, "Deformed Graph Laplacian for Semisupervised Learning," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 26, no. 10, pp. 2261–2274, 2015.
- [29] C. Xu, D. Tao, and C. Xu, "Large-Margin Multi-Label Causal Feature Learning," in *Proceedings of the 29th Conference on Artificial Intelligence (AAAI '15)*, pp. 1924–1930, 2015.
- [30] T. Liu, D. Tao, M. Song, and S. J. Maybank, "Algorithm-dependent generalization bounds for multi-task learning," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 2, article A4, pp. 227–241, 2017.
- [31] C. J. C. Burges, "A tutorial on support vector machines for pattern recognition," *Data Mining and Knowledge Discovery*, vol. 2, no. 2, pp. 121–167, 1998.
- [32] A. J. Smola and B. Schölkopf, "A tutorial on support vector regression," *Statistics and Computing*, vol. 14, no. 3, pp. 199–222, 2004.
- [33] B. Schölkopf, A. Smola, and K.-R. Müller, "Nonlinear component analysis as a kernel eigenvalue problem," *Neural Computation*, vol. 10, no. 5, pp. 1299–1319, 1998.
- [34] Y. Xu, D. Zhang, F. Song, J.-Y. Yang, Z. Jing, and M. Li, "A method for speeding up feature extraction based on KPCA," *Neurocomputing*, vol. 70, no. 4–6, pp. 1056–1061, 2007.
- [35] S. Mika, G. Ratsch, J. Weston, B. Schölkopf, and K.-R. Müller, "Fisher discriminant analysis with kernels," in *Proceedings of the 9th IEEE Signal Processing Society Workshop on Neural Networks for Signal Processing (NNSP '99)*, pp. 41–48, Madison, Wis, USA, August 1999.
- [36] K. Yu, L. Ji, and X. Zhang, "Kernel nearest-neighbor algorithm," *Neural Processing Letters*, vol. 15, no. 2, pp. 147–156, 2002.
- [37] L. Zhang, W.-D. Zhou, P.-C. Chang et al., "Kernel sparse representation-based classifier," *IEEE Transactions on Signal Processing*, vol. 60, no. 4, pp. 1684–1695, 2012.
- [38] Y. Xu, Z. Fan, and Q. Zhu, "Feature space-based human face image representation and recognition," *Optical Engineering*, vol. 51, no. 1, Article ID 017205, 2012.
- [39] K.-R. Müller, S. Mika, G. Ratsch, K. Tsuda, and B. Schölkopf, "An introduction to kernel-based learning algorithms," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 12, no. 2, pp. 181–201, 2001.
- [40] M. Kim, "Accelerated max-margin multiple kernel learning," *Applied Intelligence*, vol. 38, no. 1, pp. 45–57, 2013.
- [41] C. Xu, T. Liu, D. Tao, and C. Xu, "Local Rademacher complexity for multi-label learning," *IEEE Transactions on Image Processing*, vol. 25, no. 3, pp. 1495–1507, 2016.
- [42] K. Petersen and M. Pedersen, *The Matrix Cookbook*, 2006, <http://matrixcookbook.com/>.
- [43] Y. Xu, X. Z. Fang, X. L. Li, J. Yang, J. You, and H. Liu, "Data uncertainty in face recognition," *IEEE Transactions on Cybernetics*, 2014.

Research Article

Mining Community-Level Influence in Microblogging Network: A Case Study on Sina Weibo

Yufei Liu,¹ Dechang Pi,^{1,2} and Lin Cui¹

¹College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing, Jiangsu 211106, China

²Collaborative Innovation Center of Novel Software Technology and Industrialization, Nanjing, Jiangsu 211106, China

Correspondence should be addressed to Dechang Pi; dc.pi@nuaa.edu.cn

Received 7 June 2017; Accepted 12 November 2017; Published 4 December 2017

Academic Editor: Jia Wu

Copyright © 2017 Yufei Liu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Social influence analysis is important for many social network applications, including recommendation and cybersecurity analysis. We observe that the influence of community including multiple users outweighs the individual influence. Existing models focus on the individual influence analysis, but few studies estimate the community influence that is ubiquitous in online social network. A major challenge lies in that researchers need to take into account many factors, such as user influence, social trust, and user relationship, to model community-level influence. In this paper, aiming to assess the community-level influence effectively and accurately, we formulate the problem of modeling community influence and construct a community-level influence analysis model. It first eliminates the zombie fans and then calculates the user influence. Next, it calculates the user final influence by combining the user influence and the willingness of diffusing theme information. Finally, it evaluates the community influence by comprehensively studying the user final influence, social trust, and relationship tightness between intrausers of communities. To handle real-world applications, we propose a community-level influence analysis algorithm called CIAA. Empirical studies on a real-world dataset from Sina Weibo demonstrate the superiority of the proposed model.

1. Introduction

Community-level influence analysis is an emerging problem, which can be used in many filed, for example, recommendation system [1, 2], public opinion prediction [3], and cybersecurity analysis [4]. There are many researchers who are interested in analyzing the social influence in social networks [5], but rarely assessing the influence in community level. With the rapid spread of online social networks, such as Twitter, Facebook, and Sina Weibo, large amounts of data with the real world are produced, which provide support for the social influence analysis.

How to establish an effective model for analyzing community-level influence has become an important research for online social network. Community-level influence is greater than individual-level influence, but few researchers have studied community influence. The existing studies establish various social influence analysis models [6, 7], but they just study the influence in the individual level and mostly ignore the existence of a common influence pattern from a

community that includes multiple nodes. A large number of achievements have been obtained on individual-level influence, but most of the studies are based on static statistics method [8–11], link analysis algorithms [12–14], or probabilistic models [15–17]. These studies do not consider whether the user is willing to receive or diffuse information or what the role of social trust between users is or do not remove zombie fans. However, these factors are very important for analyzing the social influence. Meanwhile, the existing works about community-level influence focus on the influence strength between communities and ignore the problem of analyzing the community-level influence. For example, Belák et al. [18] calculated the community-level influence by only averaging influence of all users in a community.

An important observation is that zombie fans have no contribution to the social influence, and the willingness of users to diffuse information has a certain effect on the accuracy of calculating social influence, and social trust plays an important role in social influence. The trust degree of user A to user B determines the influence of user B on user A. The

more the user A trusts user B, the more influence the user B has on the user A. Because user influence is the basis of the community influence, a little carelessness on the former will lead to errors on the later.

Aiming to assess the community-level influence effectively and accurately, we construct a community-level influence analysis model that can assess community influence. Based on our model, a community-level influence analysis algorithm (short for CIAA) is proposed, which can assess the community influence more effectively and accurately. The main idea of our model is as follows. First, we eliminate the interference of zombie fans on the social influence to make the results more accurate. Then, in the process of calculating user influence, we consider the social trust and use the random walk method to calculate the user influence. In evaluating the user's theme information, the user mean willingness is calculated by exploring the content related to the user's theme information. We combine these two factors (the user influence and the user willingness to diffuse theme information) to calculate the user final influence. Finally, the community-level influence is calculated by comprehensively studying the user final influence, the social trust, and relationship tightness between intrasusers of communities. Experiments are conducted on a real-world dataset crawled from Sina Weibo. Comparing with the state-of-the-art algorithm (the averaging user influence algorithm [18]), the results show that our model is more effective and accurate to evaluate the community-level influence.

The contributions of this paper can be summarized as follows. (1) We formulate the problem of analyzing the community-level influence and design a community-level influence analysis model. (2) CIAA, a community-level influence analysis algorithm based on our model, is proposed, which is effective and reliable to evaluate the community influence of microbloggers from Sina Weibo. (3) We conduct extensive experiments to assess the performance of the proposed model. Experimental results on the real-world dataset demonstrate the superiority of the proposed CIAA.

The rest of the paper is organized as follows. In Section 2, we summarize the related works. In Section 3, we propose the community-level influence analysis model and give an example to illustrate its working principle, and the CIAA is proposed. In Section 4, we conduct experiments on the real-world dataset crawled from Sina Weibo and then analyze the performance of the proposed approach. Finally, we state the conclusion and future work in Section 5.

2. Related Works

Since Katz and Lazarsfeld [19] found that social influence plays an important role in social life and decision-making in the 1950s, researchers in computer field have spare no effort to study the relevant problems. It is found that the popular users play an important role in adopting innovation, social public opinion propagation and guidance, group behavior formation and development [5], and so on.

There are a great deal of research efforts to measure individual-level influence [20, 21], typically, the "opinion leaders." Existing methods can be categorized into three

types: the network structure based methods, the user behavior based methods, and the mutual information based methods. The network structure based methods are degree centrality [22], closeness centrality [23], betweenness centrality [24], eigenvector centrality [25], Katz centrality [26], PageRank [27], and clustering coefficient [28]. We know that node degree essentially means the connection between a node and its neighbors. The method based on node degree can intuitively express this meaning, and its computational cost is smaller than other methods [29]. These methods are widely used in measuring the users' influence in the social network. However, the methods based on node degree only reflect the connection between the users and their neighbors and cannot measure the users' influence in the entire social network for the local influence of users. For example, based on the community scale-sensitive maxdegree, Hao et al. [30] proposed an influential users discovering approach called CSSM when placing advertisements. CSSM uses the degree centrality and neighbor's degree to evaluate node's (microbloggers) influence. However, the algorithm does not consider the contribution of microblogs to user influence. Comparing with the methods based on the degree, the method based on the shortest path (closeness centrality and betweenness centrality) can measure the individual-level influence in the entire social network. Nevertheless, its computational complexity is higher than the degree centrality method. For example, based on text mining and social network analysis, Bodendorf and Kaiser [31] proposed an approach to detect opinion leaders in directed graph of user communication relationship. It can predict tendency of network opinion leaders via closeness centrality and betweenness centrality. Moreover, measuring the individual-level influence by the shortest path is an ideal status, and it is difficult to achieve in the real-world application scenarios. Besides, the methods based on random walk only consider the structure characteristics of the node while ignoring the behavior characteristics. For example, Xiang et al. [32] provided an understanding of PageRank and authority from an influence propagation perspective by performing random walks. However, they did not consider the personal attributes to understanding of PageRank as well as the relationship between PageRank and social influence analysis. Zhu et al. [33] proposed a novel information diffusion model called CTMC-ICM, which introduces the continuous-time Markov Chain theory into the Independent Cascade Model. Based on the model, they proposed a new ranking metric called SpreadRank. Based on continuous-time Markov process, Li et al. [34] proposed a dynamic information propagation model called IDM-CTMP to predict the influence dynamics of social network users. IDM-CTMP defined two other dynamic influence metrics and could predict the spreading coverage of a user within a given time period. Zhou et al. [35] established new upper bounds to significantly reduce the number of Monte-Carlo simulations in greedy-based algorithms, especially at the initial step. Based on the bound, they proposed a new upper bound based lazy forward algorithm for discovering the top- k influential nodes in social networks.

The aforementioned models focus only on assessing the social influence of single individuals. However, a small

number of works attempt to build models on the community influence analysis. Qi et al. [36] applied degree centrality, closeness centrality, and betweenness centrality to groups and classes as well as individuals. Latora and Marchiori [37] put forward a group information centrality to measure the importance of node sets. Mehmood et al. [38] exploited information diffusion records to calculate the influence strength between different communities. Although these works preliminarily study the community-level influence, none of them focuses on how to measure a community's influence. Belák et al. [18] assessed the community-level influence according to the average of the all users' influence in the same community. Because the distribution of the users' influence is uneven in different communities, average based method is inequitable to bigger communities, while summation based method is inequitable to smaller ones. At present, community-level influence analysis is still a challenging problem.

3. Proposed Methodology

We construct our model and implement the corresponding algorithm in this section. First, we give the related definitions in Section 3.1. Then, we propose the community-level influence analysis model for microbloggers. Next, we describe the working principle of our model via an example in Section 3.2. Finally, the community-level influence analysis algorithm is proposed in Section 3.3.

3.1. Related Definitions and Community-Level Influence

Analysis Model

3.1.1. Related Definitions. Social networks and communities are described as follows: a typical social network can be represented as a bipartite graph $G = \{V, E\}$, V is a set of nodes (users) in a social network, and E is a set of edges used to describe the relationships between nodes. A community can be represented as a subgraph of a social network: that is, $C = \{CV, CE\}$; $CV \subseteq V$ is a set of users in a community. $CE \subseteq E$ is a set of relationships between users within a community. A node is defined as a user within the community if he/she belongs to the community; otherwise, he/she is defined as a user outside the community. The set of users outside the community is written as UOC. Modeling and calculating the community influence of C_i are the basis of our work, and the objective function of our model is as follows:

$$CI(C_i) = f(G, C_i). \quad (1)$$

$CI(C_i)$ denotes the community influence of the community C_i , and the function $f(G, C_i)$ indicates that the assessment method is based on G and C_i . There are two entities (i.e., users and communities) which can produce influence. To study the community-level influence, we give the related definitions as follows.

Definition 1.

Trust. A node in a social network has a certain trust degree in other nodes according to its past contact with other nodes or the reputation of other nodes [39, 40]. According to the

different sources of trust, we divide the trust into direct trust and indirect trust.

(1) **Direct Trust (DT).** Assume that the node v is the entry node of the node u , indicating that there is contact between u and v . According to the previous contacts and the reputation of u , v will have direct trust on u .

(2) **Indirect Trust (IT).** Assume that the node u is the reachable node of the node v ; v will have indirect trust on u because the reputation of u can be transmitted to v .

Users not only have mutual trust, but also mutually influence each other. According to the different sources of influence, this paper divides the influence into direct influence and indirect influence.

Definition 2.

(1) **Direct Influence (DI).** Assume that the node v is the entry node of the node u ; u will have an influence on v : that is, u produces direct influence on v .

(2) **Indirect Influence (II).** Assume that the node u is a reachable node of the node v ; u will have an influence on v through transmission layer by layer: that is, u produces indirect influence on v .

In order to assess the overall influence of u on v , we define the user combined influence.

Definition 3.

User Combined Influence (UCI). Because v has direct trust or indirect trust to u , and u has direct influence or indirect influence on v , we comprehensively combine the four factors to calculate the combined influence of u on v .

Definition 4.

(1) **User Influence (UI).** User influence refers to the influence of individual on other users.

(2) **Community Influence (CI).** Community influence is the overall influence of the community, which is formed by the UI of all the users in the community and the community's self-factors.

Definition 5.

Mean Willingness to Diffuse Theme Information (MW). In communities, some users receiving the theme information may not diffuse it, some users prefer to post their own blog, and some users prefer to forward others' blog. We assess the community influence by taking into account the diffusion of information between users. MW represents a user's willingness to diffuse the information of a blog. The theme information of the user u is stored in the set $T(u) = \{t_{u1}, t_{u2}, \dots, t_{uj}, \dots\}$, where t_{uj} represents the user's j th theme information. If t_{uj} is diffused in a social network, a path map g_{uj} is formed to describe the propagation path. We store the path graphs formed by $T(u)$ in the set $g(u) = \{g_{u1}, g_{u2}, \dots, g_{uj}, \dots\}$.

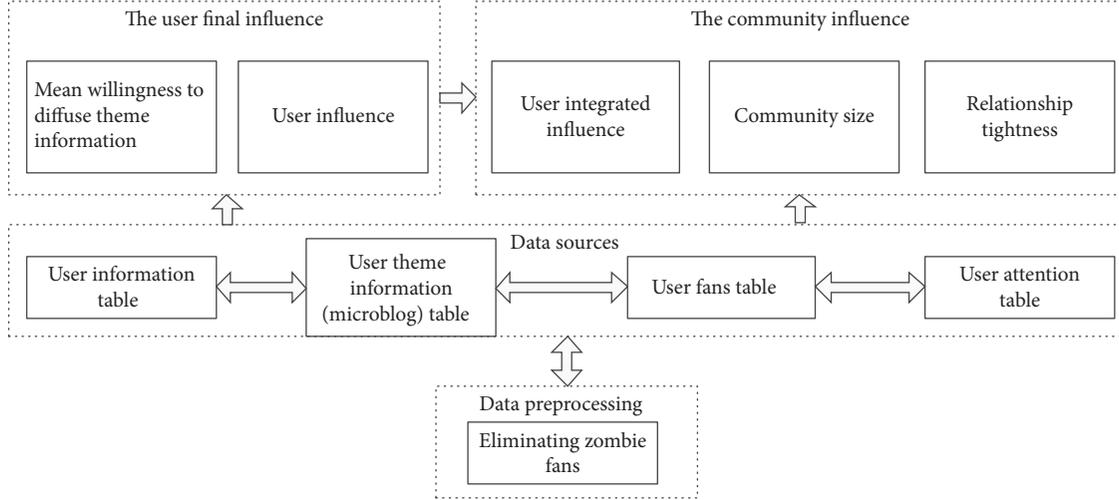


FIGURE 1: The framework of the proposed model.

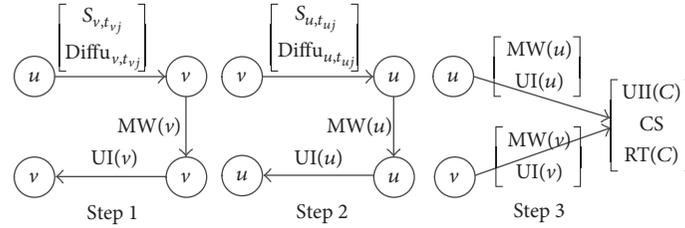


FIGURE 2: The working steps of the community-level influence analysis model.

3.1.2. Model Framework. Our model consists of four modules: data preprocessing module, data source module, the user final influence module, and the community influence module. Figure 1 shows our model framework.

Data preprocessing module is used to eliminate zombie fans. We judge the zombie fans from the behavior dimension and time dimension. Behavior dimension is based on the amount of theme information posted by the user and the fans' influence of the user. Time dimension is based on the user login frequency and the frequency of diffusing theme information. Finally, the data preprocessing results are stored to the data source.

Data source module is responsible for providing the relevant data needed for influence analysis. We establish the user information table, the microblog table, the user fans information table, and the user attention table to access the user's relevant information efficiently.

The user final influence module first calculates the mean willingness to diffuse theme information for each user in a community and then calculates the user's influence. Next, it combines these two results to get the user final influence.

The community influence module first calculates the community size, the tightness of user relationship, and the user-integrated influence in the community and then evaluates the community influence by integrating the three factors.

3.2. Working Principle. In this subsection, we introduce the working principle of each module in the model framework in

detail. We assume that u and v are two users in community C . After performing data preprocessing, Figure 2 shows the working principle, where the mathematical notations will be described in the following subsections in detail.

The working principle can be described as the following steps.

Step 1. Calculate the $Diffu_v$ and S_v of v . Then calculate the $MW(v)$ of v . Finally, calculate $UI(v)$ of v .

Step 2. According to Step 1, calculate the $MW(v)$ and $UI(v)$ of u .

Step 3. Integrate MW and UI to calculate the $UII(C)$. Then calculate CS and $RT(C)$. Finally, combine the three factors to calculate the community influence.

3.2.1. Data Preprocessing. In microblogging networks, some users of ulterior motives or business purpose lead to producing the zombie fans. According to the definition in [41], zombie fans are the users who are fake fans generated and maintained mostly for economic purpose. Zombie fans certainly interfere in analyzing the social influence. A small number of empirical researches have been conducted on recognizing zombie fans [41–43]. The existing studies were mostly subject to the Twitter platform.

Presently, researchers generally detect the zombie fans based on the amount of attention, the number of fans,

- (1) **Input:** $V, E, LF, DAF, NUI, NAU, NUF$
- (2) **Output:** $G = (V, E)$
- (3) Select the users who are the last 10% of the login frequency and whose login time interval is greater than 7 days, into the set LF
- (4) Put the users with the top 10% of the diffusing advertisement frequency into the set DAF
- (5) Select the users who are the last 10% of the number of user' theme information into the set NUI
- (6) Put the users with the top 10% of the attention users into the set NAU
- (7) Put the users with the number of fans between 10–200 into the set NUF
- (8) $ZF = LF \cap DAF \cap NUI \cap NAU \cap NUF$
- (9) Update $V = V - ZF$ and $E = E - E_{ZF}$
- (10) **return** V, E

ALGORITHM 1: Eliminating zombie fans.

original and forward information frequencies, and other basic attributes. With the ever-changing escalation of zombie fans, zombie fans will produce more features [44]. The existing feature-based methods to eliminate zombies may gradually fail. We observe that because zombie fans are occasionally managed via software program or a few people behind, zombie fans often rarely speak, even seldom log in, or no longer are used; and their behaviors can be vastly different with ordinary users in profile information and contents. Moreover, no matter how the features of zombie fans change, they can be split into time dimension and behavior dimension. Thus, it is reasonable to recognize zombie fans from the time dimension and behavior dimension, and it is more able to adapt to the needs of detecting zombie fans in microblogging networks.

According to expert knowledge criteria [45], in the time dimension, we assess zombie fans from the user login frequency and the diffusing advertisement frequency. Thus, time dimension includes login frequency (LF) and diffusing advertisement frequency (DAF). Login frequency refers to the number of logins in a period. The lower the frequency of login is, the higher the probability of the user becoming zombie fans is. The login frequency is calculated as follows:

$$LF = \frac{\Delta t \cdot \text{LoginNumber}}{\Delta t}, \quad (2)$$

where LoginNumber indicates the number of logins. The higher the diffusing advertisement frequency is, the higher the probability of the user becoming zombie fans is. The diffusing advertisement frequency is calculated as follows:

$$DAF = \frac{\Delta t \cdot \text{NumberOfDiffusingAdvertisement}}{\Delta t}, \quad (3)$$

where NumberOfDiffusingAdvertisement indicates the number of diffusing advertisement frequencies.

For the same reason, in the behavior dimension, we assess zombie fans from the amount of user theme information and the individual influence of the user's fans. Thus, we take into account the number of user theme information (NUI), the number of attention users (NAU), and the number of user's fans (NUF).

To ensure that the criteria of the parameters are reliable, the corresponding criteria are obtained by prior knowledge, expert knowledge, or experimental trial. For example, we select the users who are the last 10% of the login frequency and whose login time interval is greater than 7 days into the set LF. To reduce the amount of calculation, we filter all users in a microblogging network. If a user has a certified user in his/her fans, the user is not considered a zombie fan. If a user does not have a certified user in his/her fans, the details to eliminate zombie fans can be described in Algorithm 1.

As we can see that, unlike the classification and pattern recognition, the proposed method to eliminating zombie fans does not require labeled data and training model. It is effective and easy to use in practice.

3.2.2. The User Final Influence. The traditional models are simple, not taking into account the degree of social trust between users and the user's willingness to diffuse theme information. However, the two factors are important to the user final influence. In this paper, the user final influence is calculated by integrating the MW and UI. Because the influence of a user on other users is related to the user's willingness to exert his/her influence, the bigger the value of MW, the greater the probability of the user diffusing a theme information. UFI is calculated as follows:

$$UFI(u) = MW(u) \times UI(u). \quad (4)$$

Mean Willingness to Diffuse Theme Information. The higher frequency of diffusing theme information means a higher user influence, because more users will know the user. Therefore, MW reflects the probability that a user has high-impact in a microblogging network. The parameter $S_{v, T_{uj}}$ indicates the state of receiving theme information for the user v as follows:

$$S_{v, T_{uj}} = \begin{cases} 0, & \text{The user has never received the theme information} \\ 1, & \text{The user has received the theme information.} \end{cases} \quad (5)$$

The initial value of $S_{v, T_{uj}}$ is set to 0. Meanwhile, to know the result of v diffusing the theme information t_{uj} , we observe

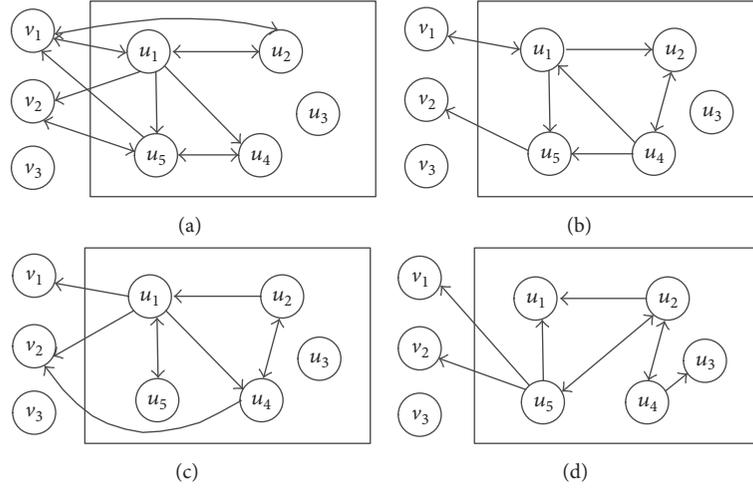


FIGURE 3: An example of calculating MW: there are five users inside a community, that is, $u_1, u_2, u_3, u_4,$ and u_5 . There are three users outside the community, that is, $v_1, v_2,$ and v_3 . (a) shows the relationship between these users. (b) shows the diffusion of theme information from u_1 . (c) also shows the diffusion of theme information from u_1 . (d) shows the diffusion of theme information from u_2 .

g_{uj} . The parameter $\text{Diffu}_{v,t_{ij}}$ indicates whether v diffuses the theme information that he/she received.

$$\text{Diffu}_{v,t_{ij}} = \begin{cases} 0, & \text{outdegree} \leq 0 \\ 1, & \text{others.} \end{cases} \quad (6)$$

When the outdegree of v is greater than 0, it indicates that v has already diffused the theme information; otherwise, v has never diffused the theme information. The number of users receiving theme information is written as NRTI and the number of users diffusing theme information is written as NDTI.

$$\begin{aligned} \text{NRTI} &= \sum_{u \in (V - \{v\})} \sum_{t_{ij} \in T(u)} S_{v,t_{ij}}, \\ \text{NDTI}_v &= \sum_{u \in (V - \{v\})} \sum_{t_{ij} \in T(u)} \text{Diffu}_{v,t_{ij}}. \end{aligned} \quad (7)$$

MW is calculated as

$$\begin{aligned} \text{MW}(v) &= \frac{\theta \times (\text{NDTI}_v / \text{NRTI}_v) + (1 - \theta) \times \sum_{u \in \text{In}(v)} \text{MW}(u) \times w(u)}{\text{num}_S} \\ &+ \frac{|\text{NP}(v)|}{\text{num}_S}, \end{aligned} \quad (8)$$

where $w(u) = 1/\text{outdegree}(u)$. $\text{MW}(v)$ is the MW of v . $\theta \in [0, 1]$ is the weight. $\text{NP}(v)$ represents the total number of theme information posts by v . $\text{In}(v)$ is the set of indegree nodes of v . $w(u)$ represents the weight of the user u , which is determined by his/her outdegree. num_S is the total number of g_{uj} . The initial value of $\text{MW}(v)$ is set as 1. We give an example for calculating MW in Figure 3.

Assume that the MW of all users initially are 1, $\theta = 0.6$, and then calculate the MW as follows.

(1) $\text{MW}(u_1)$. From Figures 3(b)–3(d), we have $\text{num}_S = 3$. For u_1 , he/she posts two-theme information, which forms two theme information graphs in Figures 3(b) and 3(c). Thus, we get the set $T(u_1)$ ($|T(u_1)| = 2$). From Figure 3(d), $\text{NRTI}_{u_1} = 1$, $\text{NDTI} = 0$, because the outdegree of node u_1 is 0, and u_1 forms its one theme information graph. The $\text{MW}(u_1)$ is calculated as follows:

$$\begin{aligned} A(u_1) &= \{u_2, u_5\}, \\ B(u_1) &= \emptyset; \\ w(u_2) &= \frac{1}{2}, \\ w(u_5) &= \frac{1}{4}, \\ \text{MW}(u_1) &= \frac{0.6 \times (0/1) + 0.4 \times (1 \times (1/2) + 1 \times (1/4))}{3} + \frac{2}{3} \\ &= \frac{23}{30}. \end{aligned} \quad (9)$$

(2) $\text{MW}(u_2)$. Similar to the calculation of $\text{MW}(u_2)$, we have the set $T(u_2)$, $|T(u_2)| = 1$. From Figures 3(b) and 3(c), we have $\text{NDTI}_{u_2} = 1$, $\text{NRTI}_{u_2} = 2$. $\text{MW}(u_2)$ is calculated as follows:

$$\begin{aligned} A(u_2) &= \{u_1, u_4\}, \\ B(u_2) &= \emptyset; \\ w(u_2) &= 1, \\ w(u_4) &= \frac{1}{3}, \\ \text{MW}(u_2) &= \frac{0.6 \times (1/2) + 0.4 \times (1 \times 1 + 1 \times (1/3))}{3} \\ &+ \frac{1}{3} = \frac{1}{18}. \end{aligned} \quad (10)$$

Similarly, for u_3 , u_4 , and u_5 , we have

$$\text{NDTI}_{u_3} = 0 + 0 + 0 = 0,$$

$$\text{NRTI}_{u_3} = 0 + 0 + 1 = 1,$$

$$\text{MW}(u_3) = \frac{0.6 \times 0 + 0.4 \times 0}{3} + 0 = 0,$$

$$\text{NDTI}_{u_4} = 1 + 1 + 1 = 3,$$

$$\text{NRTI}_{u_4} = 1 + 1 + 1 = 3,$$

$$A(u_4) = \{u_2\},$$

$$B(u_4) = \emptyset,$$

$$w(u_2) = 1,$$

$$A(u_4) = \{u_1, u_2\},$$

$$B(u_4) = \{v_2\},$$

$$w(u_1) = \frac{1}{3},$$

$$w(u_2) = \frac{1}{2},$$

$$w(v_2) = 1,$$

$$A(u_4) = \{u_2\},$$

$$B(u_4) = \emptyset,$$

$$w(u_2) = \frac{1}{3},$$

$$\text{MW}(u_4) = \frac{0.6 \times (3/3) + 0.4 \times (1 \times 1 + 1 \times (1/3) + 1 \times (1/2) + 1 \times 1 + 1 \times (1/3))}{3} + 0 = \frac{28}{65},$$

$$\text{NDTI}_{u_5} = 0 + 1 + 1 = 2,$$

$$\text{NRTI}_{u_5} = 1 + 1 + 1 = 3,$$

$$A(u_5) = \{u_1\},$$

$$B(u_5) = \emptyset,$$

$$w(u_2) = \frac{1}{3},$$

$$w(u_4) = \frac{1}{3},$$

$$A(u_5) = \{u_1\},$$

$$B(u_5) = \emptyset,$$

$$w(u_1) = \frac{1}{3},$$

$$A(u_5) = \{u_2\},$$

$$B(u_5) = \{v_2\},$$

$$\begin{aligned}
w(u_2) &= \frac{1}{3}, \\
w(v_2) &= 1, \\
MW(u_5) &= \frac{0.6 \times (2/3) + 0.4 \times (1 \times (1/3) + 1 \times 1)}{3} + 0 = \frac{4}{9}.
\end{aligned} \tag{11}$$

The User Influence. There are mutual impact and mutual trust between users. Social trust plays an important role in calculating the user influence. She/he is impacted by others including users inside and outside the community.

(1) *Calculating Direct Trust and Direct Influence.* If v is an entry node of u , then v will have direct trust on u .

$$\begin{aligned}
DT_{vu} &= \frac{RU(u)}{\text{outdegree}(v)}, \\
RU(u) &= \frac{\sum_{w \in \text{In}(u)} RU(w)}{\text{indegree}(u)},
\end{aligned} \tag{12}$$

where DT_{vu} is the direct trust of v on u . $RU(u)$ is the reputation of user u . $\text{In}(u)$ is the set of entry nodes of u , and $RU(u \leftarrow w)$ is the reputation of the entry neighbor w of u . The value of $RU(u)$ depends on the average reputation of all u 's entry neighbors. For each node, we give the initial direct trust value 0.1. In Figure 3(a), we calculate the direct trust on u_1 from other nodes as follows:

$$\begin{aligned}
RU(u_1) &= \frac{0.1 + 0.1 + 0.1 + 0.1}{4 + 1} = 0.08, \\
\text{In}(u_1) &= \{u_2, u_4, u_5, v_1\}, \\
DT_{u_2, u_1} &= \frac{0.08}{2} = 0.04, \\
DT_{u_3, u_1} &= \frac{0.08}{0} \text{ (written as 0)}, \\
DT_{u_4, u_1} &= \frac{0.08}{2} = 0.04, \\
DT_{u_5, u_1} &= \frac{0.08}{4} = 0.02, \\
DT_{v_1, u_1} &= \frac{0.08}{2} = 0.04, \\
DT_{v_2, u_1} &= \frac{0.08}{1} = 0.08, \\
DT_{v_3, u_1} &= \frac{0.08}{0} \text{ (written as 0)}.
\end{aligned} \tag{13}$$

u has a direct influence on v as follows:

$$\begin{aligned}
DI_{uv} &= \frac{I(u \leftarrow v)}{\text{outdegree}(v)}, \\
W(u \leftarrow v) &= \frac{|\text{theme}(v, u)|}{\text{NRTI}_v},
\end{aligned} \tag{14}$$

where DI_{uv} is the direct influence of u on v . $I(u \leftarrow v)$ is the degree of interest of v to u . $|\text{theme}(v, u)|$ is the amount of the theme information from u in the receiving theme information of v .

In Figure 3, we calculate the direct influence on u_1 produced by other users as follows:

$$\begin{aligned}
I(u_1 \leftarrow u_2) &= \frac{2}{2} = 1, \\
I(u_1 \leftarrow u_3) &= \frac{0}{1} = 0, \\
I(u_1 \leftarrow u_4) &= \frac{2}{3} = 0.667, \\
I(u_1 \leftarrow u_5) &= \frac{2}{3} = 0.667, \\
I(u_1 \leftarrow v_1) &= \frac{2}{3} = 0.667, \\
I(u_1 \leftarrow v_2) &= \frac{2}{3} = 0.667, \\
I(u_1 \leftarrow u_3) &= \frac{2}{0} \text{ (written as 0)}.
\end{aligned} \tag{15}$$

In Figure 3(a), we have

$$\begin{aligned}
DI_{u_1 u_2} &= \frac{1}{2} = 0.5, \\
DI_{u_1 u_3} &= \frac{0}{0} \text{ is 0}, \\
DI_{u_1 u_4} &= \frac{0.667}{2} = 0.334, \\
DI_{u_1 u_5} &= \frac{0.667}{5} = 0.133, \\
DI_{u_1 v_1} &= \frac{0.667}{2} = 0.334, \\
DI_{u_1 v_2} &= \frac{1}{1} = 1, \\
DI_{u_1 v_3} &= \frac{0}{0} \text{ (written as 0)}.
\end{aligned} \tag{16}$$

(2) *Indirect Trust and Indirect Influence.* If u is the reachable node of v , then v will have indirect trust on u as follows:

$$IT_{vu} = \frac{RU(u)}{\min_{vu}}. \tag{17}$$

IT_{vu} is v 's indirect trust on u . \min_{vu} is the length of the shortest path from v to u .

In Figure 3(a), we calculate the indirect trust on u_1 gained from other nodes as follows:

$$\begin{aligned}
IT_{u_2u_1} &= \frac{0.08}{1} = 0.08, \\
IT_{u_3u_1} &= \frac{0.08}{0} \text{ (written as 0)}, \\
IT_{u_4u_1} &= \frac{0.08}{1} = 0.08, \\
IT_{u_5u_1} &= \frac{0.08}{1} = 0.08, \\
IT_{v_1u_1} &= \frac{0.08}{1} = 0.08, \\
IT_{v_2u_1} &= \frac{0.08}{2} = 0.04, \\
IT_{v_3u_1} &= \frac{0.08}{0} \text{ (written as 0)}.
\end{aligned} \tag{18}$$

u has an indirect influence on v as follows:

$$\begin{aligned}
\Pi_{uv} &= \frac{I(u \leftarrow v)}{\min_{vu}}, \\
I(u \leftarrow v) &= \frac{|\text{theme}(v, u)|}{\text{NRTI}_v}.
\end{aligned} \tag{19}$$

In Figure 3(a), we calculate the indirect influence of other nodes on u_1 as follows. The calculation of I is the same as the above formula.

$$\begin{aligned}
\Pi_{u_1u_2} &= \frac{1}{1} = 1, \\
\Pi_{u_1u_3} &= \frac{0}{0} \text{ (written as 0)}, \\
\Pi_{u_1u_4} &= \frac{0.667}{1} = 0.667, \\
\Pi_{u_1u_5} &= \frac{0.667}{1} = 0.667, \\
\Pi_{u_1v_1} &= \frac{0.667}{1} = 0.667, \\
\Pi_{u_1v_2} &= \frac{1}{2} = 0.5, \\
\Pi_{u_1v_3} &= \frac{0}{0} \text{ (written as 0)}.
\end{aligned} \tag{20}$$

(3) *User Combined Influence.* Assuming that v can reach u through a path, we introduce the factor λ ($\lambda \in [0, 1]$).

If v is the entry node of u , the combined influence of u on v is

$$UCI_{uv} = \lambda DI_{uv} + (1 - \lambda) DT_{vu}. \tag{21}$$

If v is not an entry node of node u , but u is a reachable node of v , the combined influence is

$$UCI_{uv} = \lambda \Pi_{uv} + (1 - \lambda) IT_{vu}. \tag{22}$$

Assume $\lambda = 0.3$. In Figure 3, we calculate the combined influence of other nodes on u_1 as follows.

u_2 is the entry node of u_1 ; then we have $UCI_{u_1u_2} = 0.3 \times 0.5 + 0.7 \times 0.04 = 0.178$.

u_4 is the entry node u_1 ; then we have $UCI_{u_1u_4} = 0.3 \times 0.334 + 0.7 \times 0.04 = 0.1282$.

u_5 is the entry node of u_1 ; then we have $UCI_{u_1u_5} = 0.3 \times 0.133 + 0.7 \times 0.02 = 0.0539$.

v_1 is the entry node of u_1 ; then we have $UCI_{u_1v_1} = 0.3 \times 0.334 + 0.7 \times 0.04 = 0.1282$.

v_2 is the reachable node of u_1 ; then we have $UCI_{u_1v_2} = 0.3 \times 0.5 + 0.7 \times 0.04 = 0.178$.

(4) *User Influence.* User influence is got by combining all users' influence:

$$UI(u) = \frac{\sum_{v \in \text{SUCP}(u)} UCI_{uv}}{|\text{SUCP}(u)|}, \tag{23}$$

where SUCP represents a set of users that can reach u through a certain path. For example, in Figure 3, the user influence of u_1 is calculated as follows:

$$\begin{aligned}
UI(u_1) &= \frac{UCI_{u_1u_2} + UCI_{u_1u_4} + UCI_{u_1u_5} + UCI_{u_1v_1} + UCI_{u_1v_2}}{5} \\
&= 0.133.
\end{aligned} \tag{24}$$

When we get $MW(u_1)$ and $UI(u_1)$, the user final influence can be calculated according to (4).

3.2.3. *Community Influence.* The community influence is composed of the users' interaction inside and outside the community. In this paper, we consider it from three factors, that is, the user-integrated influence, the community size, and the degree of relationship tightness among users inside the community.

User-integrated influence (UII) is integrated from the final influence of all users within the community.

$$UII(C_i) = \sum_{u \in CV(u)} UFI(u), \tag{25}$$

where $UII(C_i)$ is UII of the community C_i . $CV(u)$ is the set of users inside community C_i .

The community size (CS) is important to the calculation of the community-level influence. The larger the number of users in a community is, the greater the influence of the community becomes. The formula is as follows:

$$CS(C_i) = \frac{|CV(C_i)|}{\max(V)}, \tag{26}$$

where $|CV(C_i)|$ represents the number of users in a community and $\max(V)$ represents the total number of users in the social network.

```

Input:  $G = \{V, E\}; C; T(u); g(u); \text{UII} = 0; \tau; \rho; \text{RT} = 0$ 
Output: community influence
(1) for  $i = 0$  to  $|V|$  do
(2)      $\text{MW}(i)$ 
(3)      $\text{UI}(i)$ 
(4) end for
(5) for  $j = 0$  to  $|CV|$  do
(6)      $\text{UII}(j) = \text{MW}(j) \times \text{UI}(j) + \text{UII}(j)$ 
(7) end for
(8)  $\text{CS}(C)$ 
(9) for  $i = 0$  to  $|CV|$  do
(10)     $\text{RT}(C_i) = \frac{\sum_{u \in CV(C_i)} (\text{outdegree}(u) + \text{indegree}(u))}{CV(C_i)}$ 
(11) end for
(12)  $\text{CI}(C_i) = \tau \times \text{UII}(C_i) + \rho \times \text{CS} + (1 - \tau - \rho) \times \text{RT}(C_i)$ 
(13) return  $\text{CI}(C_i)$ 

```

ALGORITHM 2: Community-level influence analysis algorithm (CIAA).

The degree of relationship tightness (RT) represents the degree of closeness between users inside a community. We describe it from the user's outdegree and indegree as follows:

$$\text{RT}(C_i) = \frac{\sum_{u \in CV(C_i)} (\text{outdegree}(u) + \text{indegree}(u))}{CV(C_i)}. \quad (27)$$

Therefore, we calculate the CI as follows:

$$\text{CI}(C_i) = \tau \times \text{UII}(C_i) + \rho \times \text{CS} + (1 - \tau - \rho) \times \text{RT}(C_i), \quad (28)$$

where τ and ρ ($\tau, \rho \in [0, 1]$) are used to distinguish the importance of different factors.

3.3. The Proposed Algorithm. According to the above description, we propose a community-level influence analysis algorithm, called CIAA, in a pseudo-code format in Algorithm 2. It can be seen from the algorithm that the total time complexity is $O(n)$. This means that our algorithm can be applied on large-scale social dataset.

4. Experiments

We conduct experiments to validate the effectiveness of the proposed approach on a real-world microblogging network. In this section, we describe the experimental setup followed by the discussion of experiment results.

4.1. Dataset. The real-world dataset in this paper is crawled from Sina Weibo by Weibo crawler. Similar to a hybrid of Twitter and Facebook, Sina Weibo is one of the most popular sites in China. It has more than 33% of the Internet users in China, and its market penetration is equivalent to that of Twitter in the United States. As released by the Sina Weibo, as of June 2016, the active users from different social and cultural backgrounds have reached 282 million monthly and 86.8 million daily. Moreover, there are nearly 100 million new

TABLE 1: Data structure and description of the user information.

Features	Description
UserID	User' ID
IsVIP	Authenticated user
FansNum	Number of fans
AttenNum	Number of attention users
ThemeAmo	Amount of theme information
Tag	User' label
Time	Login time

TABLE 2: Data structure and description of the user theme information (microblogs).

Features	Description
ThemeID	Theme information ID
ThemeFromID	Source ID of theme information
ProNum	Number of processes
ThemeClass	Theme information class
PTime	Post time of theme information

TABLE 3: Data structure and description of the user fans.

Features	Description
UserID	User' ID
FansID	Fans' ID

microblogs every day. They promote and disseminate views and attitudes on business, culture, education, and so forth. The crawled data includes 20,151,129 microblogs, 932,578,467 comments, and 9,218 users. In this paper, we collected more than 1000 users from the crawled dataset and divided the related information into Tables 1, 2, 3, and 4 for data sources according to our model framework. They are stored in txt-formatted files.

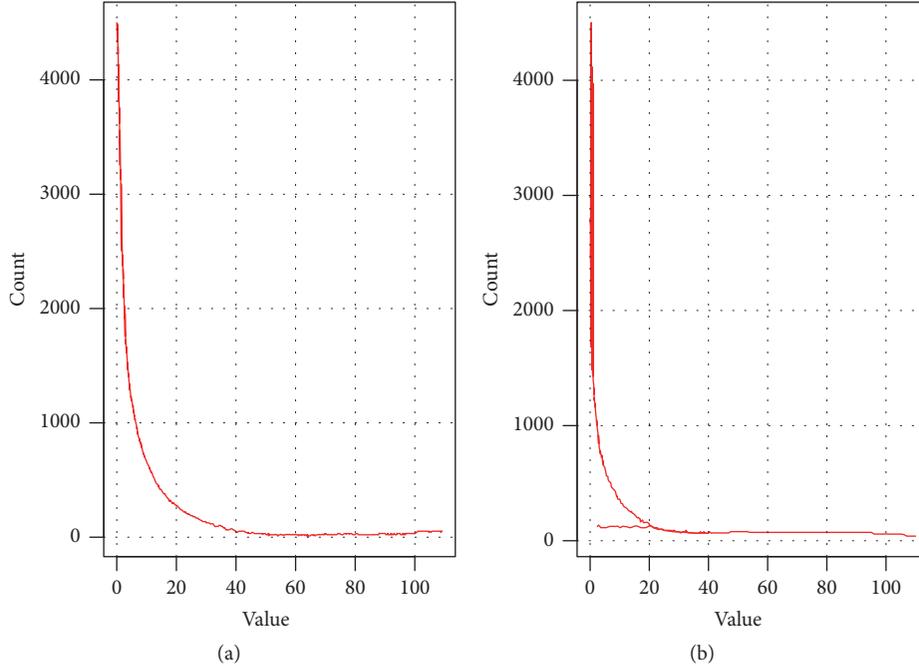


FIGURE 4: (a) is the outdegree distribution and (b) is the degree distribution.

TABLE 4: Data structure and description of the user attention.

Features	Description
UserID	User' ID
AttenID	User-attended ID

TABLE 5: Parameters for experiments.

Symbol	Description	Value
V	The total number of nodes	1127
CV	The total number of nodes in the community	20
λ	Parameter	0.3
θ	Parameter	0.5
τ	Parameter	0.5
ρ	Parameter	0.3

4.2. *Experimental Setting.* All experiments are conducted on a PC with Intel Core i5 processor, 8 GB RAM. According to prior knowledge, we set the parameters of the experiments as Table 5.

4.3. Results

4.3.1. *Community Structure Analysis.* In order to mine and study the characteristic of community, we plot the outdegree distribution and degree distribution of users in community. In a directed social network, the indegree of nodes is the number of fans of the user. The outdegree of nodes is the amount of the user's attention. Figure 4 shows the outdegree and degree distribution of data sources.

As shown in Figure 4, the outdegree distribution and the degree distribution of Sina Weibo dataset follow the power-law distribution, which indicates that the social network composed of the dataset is a scale-free network.

4.3.2. *Eliminating Zombie Fans.* In order to improve the accuracy of our model, we remove zombie fans. According to the eliminating zombie fans method in Algorithm 1, we finally remove 12 zombie fans, as shown in Table 6.

As shown in Table 6, the three sets are NUI, NAU, and NUF. The little black boxes in Table 6 represent the shared users of three sets, and they are the same as the shared users from time dimension and behavior dimension. Therefore, the shared users will be removed. We compare the user final influence without the zombie fans with the user final influence with the zombie fans, as shown in Table 7.

From Table 7, the result of the comparison shows that the accuracy of the UFI with zombie fans for the actual user ranking is only 60%. It is concluded that the elimination of zombie fans is very important for the accuracy of the user final influence.

4.3.3. *Accuracy Analysis of the User Final Influence.* We calculate the user final influence of users in community, but we compare the top ten users for simplicity. The top 10 user final influences and their related information are shown in Table 8.

According to the UFI ranking in Table 8, we find that these users are authenticated user. It is concluded that the authenticated users are more influential in microblogging networks. There are two reasons for this phenomenon. First, the majority of well-known users are authenticated users, and the influence of well-known users is larger than the user

TABLE 6: Three user sets for eliminating zombie fans. The boxes represent zombie fans.

NUF			NAU			NUI		
522	214	348	122	257	226	320	511	320
565	514	569	203	508	506	377	267	377
553	314	512	540	535	551	314	209	314
550	560	345	272	540	564	260	535	260
557	362	553	532	541	326	299	569	299
528	169	241	508	236	546	255	519	255
551	568	538	567	124	366	295	174	295
295	293	237	514	541	140	312	176	312
549	512	267	561	365	357	312	381	312
108	531	516	219	557	547	275	522	275
563	540	535	554	562	558	272	180	272
560	508	395	519	295	520	393	508	393
320	514	531	395	217	564	325	267	325
553	503	531	240	155	291	299	194	299
365	217	518	398	535	564	398	519	398
565	368	540	569	563	564	526	213	526
561	241	393	308	523	551	297	299	297
524	301	260	553	516	269	506	365	506
315	546	553	315	562	377	327	263	327
530	398	282	199	286	349	184	505	184
558	175	506	564	537	387	293	281	387
557	559	558	531	266	202	530	249	202
565	559	381	563	564	561	206	393	206
564	521	377	190	554	562	227	393	227
535	564	532	190	181	282	324	367	324
558	561	326	163	550	524	246	354	246
527	533	183	567	558	524	107	202	107
537	536	107	548	562	546	259	140	259
528	384	215	562	558	554	395	206	395
297	207	373	186	219	508	371	240	371
535	361	331	532	356	565	177	292	356
539	561	372	372	557	548	321	289	321
562	562	385	558	363	376	299	378	363
558	558	558	327	559	557	315	280	327
316	387	375	554	551	293	562	386	551
560	173	569	362	185	558	346	219	362
288	393	387	292	122	352	562	166	292
549	565	538	558	531	561	521	248	538
559	558	558	557	531	246	257	246	531

TABLE 7: Comparison of the user final influence.

User ID	UFI without zombie fans	UFI with zombie fans	The actual rankings
263 * * * *023	1	3	1
511 * * * *843	2	2	2
519 * * * *020	3	1	3
508 * * * *496	4	4	4
550 * * * *598	5	5	5
267 * * * *724	6	6	6
365 * * * *215	8	8	7
299 * * * *593	7	7	8
522 * * * *989	9	9	9
194 * * * *451	10	10	10

TABLE 8: Top 10 user information of the UFI.

UFI ranking	User ID	Number of fans	Number of blogs	Authenticated or not
1	263 * * * *023	128	1515	1
2	511 * * * *843	282	1282	1
3	519 * * * *020	66	101	1
4	508 * * * *496	261	5471	1
5	550 * * * *598	14	22	1
6	267 * * * *724	823	1452	1
7	299 * * * *593	158	109	1
8	365 * * * *215	177	945	1
9	522 * * * *989	13	29	1
10	194 * * * *451	69	11	1

average influence. Second, the authenticated user's identity is transparent, which makes the user have higher social trust. Table 8 also shows that the user final influence needs to be considered from the quality of the user fans, the number of user microblogs, and user authentication.

Table 9 and Figure 5 show the comparison between the UFI method and the microblog-fans ranking algorithm. Table 9 shows the UFI method ranking and the corresponding ranking via microblog-fans ranking algorithm. Figure 5 shows the overall ranking order via the microblog-fans ranking algorithm.

It can be seen from Table 9 and Figure 5 that the UFI ranking is almost completely different from the microblog-fans ranking. Overall, according to the UFI method, the number of microblogs and fans of the top users must reach a certain quantity to support individual influence. Thus, the number of microblogs and fans is a factor of measuring influence in UFI method. However, social trust between users can help improve individual influence in the UFI method.

The user final influence is an experimental evaluation of the user, and there is no existing dataset with its comparison. We can only refer to the ranking of the user influence from some affiliations. Based on the ranking of user influence provided by Sina Weibo official, we verify the calculation

TABLE 9: Comparison of UFI method with microblog-fans ranking algorithm.

UFI ranking	User ID	Number of fans	Number of blogs	Microblog-fans ranking
1	263 * * * *023	128	1515	3
2	511 * * * *843	282	1282	4
3	519 * * * *020	66	101	8
4	508 * * * *496	261	5471	1
5	550 * * * *598	14	22	6
6	267 * * * *724	823	1452	2
7	299 * * * *593	158	109	7
8	365 * * * *215	177	945	5
9	522 * * * *989	13	29	10
10	194 * * * *451	69	11	9

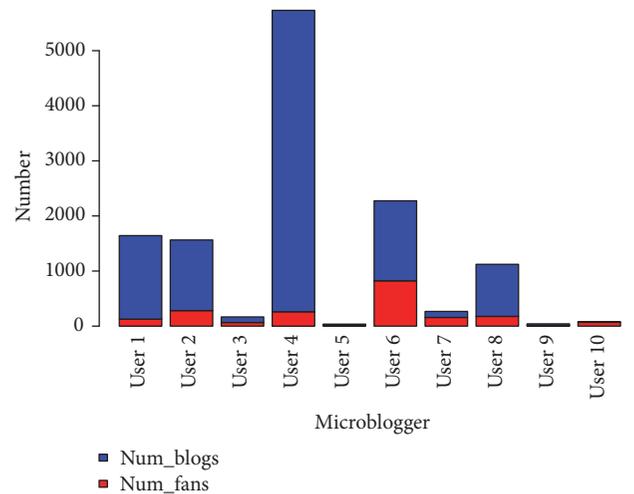


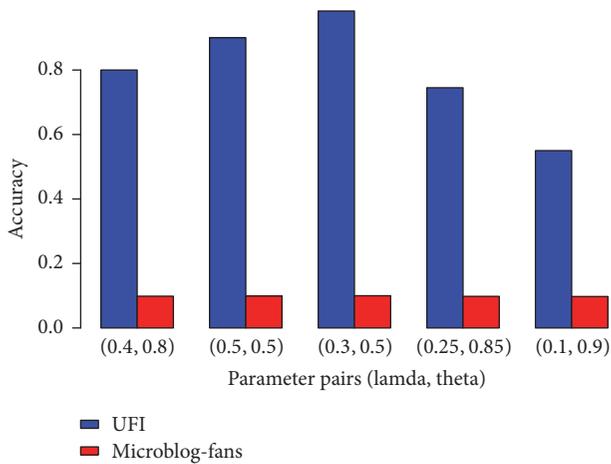
FIGURE 5: The overall ranking via the microblog-fans ranking algorithm.

method proposed in this paper. We compare the results of the proposed method with the official ranking to verify the correctness of the user final influence. Because each microblogging platform has its own influence calculation method, we cannot numerically compare the results, but we compare the results from the relative position, that is, ranking. If the influence rankings of the two methods are in the similar order, we consider the results of the influence analysis to be similar. The comparison of the users ranking by Sina Weibo officially and UFI method is shown in Table 10.

In Table 10, the user final influence calculation method and the user actual ranking are mainly the same but having the user pair of 299 * * * *593 and 365 * * * *215. That is because user influence ranking by Sina Weibo emphasizes the number of microblogs and fans, and the number of microblogs and fans of user 299 * * * *593 and user 365 * * * *215 is largely different. However, the UFI method considers the factors of influence more reasonably.

TABLE 10: Comparison of user actual ranking with UFI ranking.

User ID	The actual ranking	UFI value	UFI ranking
263 * * * *023	1	1.0000	1
511 * * * *843	2	0.0384	2
519 * * * *020	3	0.0215	3
508 * * * *496	4	0.0107	4
550 * * * *598	5	0.0099	5
267 * * * *724	6	0.00726	6
299 * * * *593	8	0.0028	7
365 * * * *215	7	0.0021	8
522 * * * *989	9	0.0019	9
194 * * * *451	10	0.0016	10

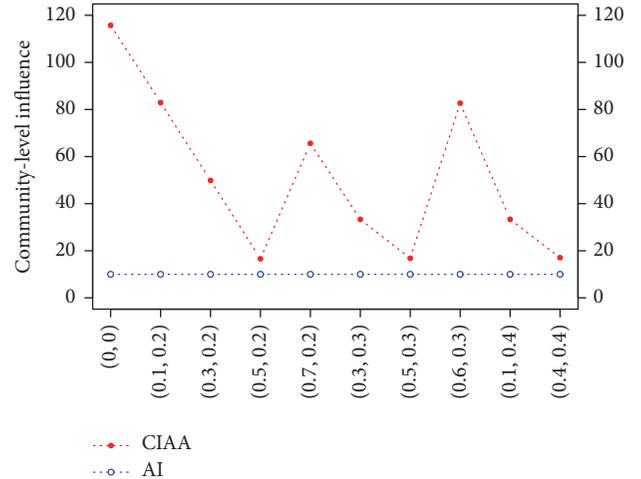
FIGURE 6: Comparison of accuracy of two methods with different λ and θ .

Considering the results of Sina Weibo official as the standard, the accuracy of UFI method will change with different λ and θ , as shown in Figure 6.

From Figure 6, it can be seen that the UFI method accuracy changes with the different λ and θ . When $\lambda = 0.3$, $\theta = 0.5$, UFI method has the highest accuracy. Therefore, the parameter pair (0.3, 0.5) is used for other experiments. We also find that the UFI method is more accurate than the microblog-fans ranking algorithm. Moreover, this experiment indicates the importance of the user willingness to diffusing theme information in the accuracy of the user influence.

4.3.4. Accuracy Analysis of CIAA. Because the existing studies of community influence are few, we compare the proposed algorithm CIAA with the averaging user influence algorithm (AI). We set different parameters pair τ and ρ for comparing the two algorithms. Then, we can calculate the corresponding community influence, as shown in Figure 7.

Figure 7 shows that the results of the CIAA are changing with the different parameter values. When $\tau = 0.5$ and $\rho = 0.2$, the results of the two algorithms are closest. That is

FIGURE 7: The community-level influence by two measuring algorithms with different (τ, ρ) pairs.

because the AI algorithm is mainly the weighted average of the user influence, and the CIAA is the integration of the user-integrated influence, the community size, and the degree of relationship tightness among users inside the community. The greater the proportion of the user final influence, the closer the results of the two algorithms. Therefore, the proposed algorithm outperforms the state-of-the-art baseline algorithm.

5. Conclusion

In this paper, we studied the emerging problem on how to model community-level influence. Online social networks, especially microblogging networks, are more and more important in our daily life. Previous works can effectively cope with the individual influence in microblogging network, but they rarely evaluate the social influence in community level, which outweighs the individual influence. We defined the related concepts for the community-level influence and constructed a model that combined the user influence, social trust, and relationship tightness of intrasurers in a community to reveal the community-level influence appropriately. We proposed the algorithm CIAA to cope with the real-world applications. We conducted empirical studies on a real-world microblogging crawled from Sina Weibo, where the CIAA outperformed the state-of-the-art baseline algorithm. To the best of our knowledge, the proposed approach has a significant effect on community influence in microblogging network. The highlights of this paper can be summarized as follows: (1) formulating the problem of analyzing community-level influence and designing a community-level influence analysis model; (2) proposing community-level influence analysis algorithm called CIAA, to cope with real-world microblogging applications; and (3) extensively demonstrating the superiority of the proposed method. In the future work, we plan to extend the proposed method to assess the community influence in dynamic online social network.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under Grants U1433116 and 61702355, in part by the Fundamental Research Funds for the Central Universities under Grant NP2017208, and in part by the Funding of Jiangsu Innovation Program for Graduate Education under Grants KYLX15_0324 and KYLX15_0321.

References

- [1] L. Yao, Q. Z. Sheng, A. H. H. Ngu, J. Yu, and A. Segev, "Unified collaborative and content-based web service recommendation," *IEEE Transactions on Services Computing*, vol. 8, no. 3, pp. 453–466, 2015.
- [2] J. Wu, S. Pan, X. Zhu, C. Zhang, and X. Wu, "Multi-instance Learning with Discriminative Bag Mapping," *IEEE Transactions on Knowledge and Data Engineering*, pp. 1–16, 2018.
- [3] D. Ghadiyaram and A. C. Bovik, "Massive online crowdsourced study of subjective and objective picture quality," *IEEE Transactions on Image Processing*, vol. 25, no. 1, pp. 372–387, 2016.
- [4] T. Cruz, L. Rosa, J. Proenca et al., "A Cybersecurity Detection Framework for Supervisory Control and Data Acquisition Systems," *IEEE Transactions on Industrial Informatics*, vol. 12, no. 6, pp. 2236–2246, 2016.
- [5] D. Kim, D. Hyeon, J. Oh, W.-S. Han, and H. Yu, "Influence maximization based on reachability sketches in dynamic graphs," *Information Sciences*, vol. 394–395, pp. 217–231, 2017.
- [6] G. Wang, W. Jiang, J. Wu, and Z. Xiong, "Fine-grained feature-based social influence evaluation in online social networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 9, pp. 2286–2296, 2014.
- [7] J. M. Hofman, A. Sharma, and D. J. Watts, "Prediction and explanation in social systems," *Science*, vol. 355, no. 6324, pp. 486–488, 2017.
- [8] S. Myers and J. Leskovec, "The bursty dynamics of the twitter information network," in *Proceedings of the 23rd International Conference on World Wide Web, WWW 2014*, pp. 913–923, Republic of Korea, April 2014.
- [9] Y. Liu, Q. Li, X. Tang, N. Ma, and R. Tian, "Superedge prediction: What opinions will be mined based on an opinion supernetwork model?" *Decision Support Systems*, vol. 64, pp. 118–129, 2014.
- [10] R. De Caux, C. Smith, D. Kniveton, R. Black, and A. Philippides, "Dynamic, small-world social network generation through local agent interactions," *Complexity*, vol. 19, no. 6, pp. 44–53, 2014.
- [11] J. Wu, S. Pan, X. Zhu, C. Zhang, and P. S. Yu, "Multiple structure-view learning for graph classification," *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–16, 2017.
- [12] L. Zhu, D. Guo, J. Yin, G. V. Steeg, and A. Galstyan, "Scalable temporal latent space inference for link prediction in dynamic social networks," *IEEE Transactions on Knowledge and Data Engineering*, vol. 28, no. 10, pp. 2765–2777, 2016.
- [13] S.-Y. Tan, J. Wu, L. Lü, M.-J. Li, and X. Lu, "Efficient network disintegration under incomplete information: The comic effect of link prediction," *Scientific Reports*, vol. 6, Article ID 22916, 2016.
- [14] J. Wu, S. Pan, X. Zhu, C. Zhang, and X. Wu, "Positive and unlabeled multi-graph learning," *IEEE Transactions on Cybernetics*, vol. 47, no. 4, pp. 818–829, 2017.
- [15] A. Almaatouq, L. Radaelli, A. Pentland, and E. Shmueli, "Are you your friends' friend? Poor perception of friendship ties limits the ability to promote behavioral change," *PLoS ONE*, vol. 11, no. 3, Article ID e0151588, 2016.
- [16] Q. Fang, J. Sang, C. Xu, and Y. Rui, "Topic-sensitive influencer mining in interest-based social media networks via hypergraph learning," *IEEE Transactions on Multimedia*, vol. 16, no. 3, pp. 796–812, 2014.
- [17] J. Wu, S. Pan, X. Zhu, and Z. Cai, "Boosting for multi-graph classification," *IEEE Transactions on Cybernetics*, vol. 45, no. 3, pp. 416–429, 2015.
- [18] V. Belák, S. Lam, and C. Hayes, "Towards maximising cross-community information diffusion," in *Proceedings of the 2012 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, ASONAM 2012*, pp. 171–178, Turkey, August 2012.
- [19] P. F. Lazarsfeld, *Personal Influence: The Part Played by People in the Flow of Mass Communications*, Transaction Publishers, New York, NY, USA, 2006.
- [20] C. Dong, Y. Zhao, and Q. Zhang, "Assessing the influence of an individual event in complex fault spreading network based on dynamic uncertain causality graph," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 27, no. 8, pp. 1615–1630, 2016.
- [21] N. Ma and Y. Liu, "SuperedgeRank algorithm and its application in identifying opinion leader of online public opinion supernetwork," *Expert Systems with Applications*, vol. 41, no. 4, pp. 1357–1368, 2014.
- [22] X. Tang, J. Wang, J. Zhong, and Y. Pan, "Predicting essential proteins based on weighted degree centrality," *IEEE Transactions on Computational Biology and Bioinformatics*, vol. 11, no. 2, pp. 407–418, 2014.
- [23] M. K. Tarkowski, P. Szczepa, T. Rahwan, T. P. Michalak, and M. Wooldridge, "Closeness centrality for networks with overlapping community structure," in *Proceedings of the in Thirtieth AAAI Conference on Artificial Intelligence*, pp. 622–629, Phoenix, Ariz, USA, 2016.
- [24] N. Kourtellis, G. De Francisci Morales, and F. Bonchi, "Scalable Online Betweenness Centrality in Evolving Graphs," *IEEE Transactions on Knowledge and Data Engineering*, vol. 27, no. 9, pp. 2494–2506, 2015.
- [25] G. Lohmann, D. S. Margulies, A. Horstmann et al., "Eigenvector centrality mapping for analyzing connectivity patterns in fMRI data of the human brain," *PLoS ONE*, vol. 5, no. 4, Article ID e10232, 2010.
- [26] P. Grindrod and D. J. Higham, "A matrix iteration for dynamic network summaries," *SIAM Review*, vol. 55, no. 1, pp. 118–128, 2013.
- [27] D. F. Gleich, "PageRank beyond the web," *SIAM Review*, vol. 57, no. 3, pp. 321–363, 2015.
- [28] J. Wang, M. Li, H. Wang, and Y. Pan, "Identification of essential proteins based on edge clustering coefficient," *IEEE Transactions on Computational Biology and Bioinformatics*, vol. 9, no. 4, pp. 1070–1080, 2012.
- [29] O. Sporns, "Contributions and challenges for network models in cognitive neuroscience," *Nature Neuroscience*, vol. 17, pp. 652–660, 2014.

- [30] F. Hao, M. Chen, C. Zhu, and M. Guizani, "Discovering influential users in micro-blog marketing with influence maximization mechanism," in *Proceedings of the 2012 IEEE Global Communications Conference, GLOBECOM 2012*, pp. 470–474, USA, December 2012.
- [31] F. Bodendorf and C. Kaiser, "Detecting opinion leaders and trends in online social networks," in *Proceedings of the 2nd ACM Workshop on Social Web Search and Mining, SWSM'09, Co-located with the 18th ACM International Conference on Information and Knowledge Management, CIKM 2009*, pp. 65–68, China, November 2009.
- [32] B. Xiang, Q. Liu, E. Chen, H. Xiong, Y. Zheng, and Y. Yang, "PageRank with priors: An influence propagation perspective," in *Proceedings of the 23rd International Joint Conference on Artificial Intelligence, IJCAI'13*, pp. 2740–2746, Beijing, China, 2013.
- [33] T. Zhu, B. Wang, B. Wu, and C. Zhu, "Maximizing the spread of influence ranking in social networks," *Information Sciences*, vol. 278, pp. 535–544, 2014.
- [34] J. Li, W. Peng, T. Li, T. Sun, Q. Li, and J. Xu, "Social network user influence sense-making and dynamics prediction," *Expert Systems with Applications*, vol. 41, no. 11, pp. 5115–5124, 2014.
- [35] C. Zhou, P. Zhang, W. Zang, and L. Guo, "On the upper bounds of spread for greedy algorithms in social network influence maximization," *IEEE Transactions on Knowledge and Data Engineering*, vol. 27, no. 10, pp. 2770–2783, 2015.
- [36] X. Qi, E. Fuller, R. Luo, and C.-Q. Zhang, "A novel centrality method for weighted networks based on the Kirchhoff polynomial," *Pattern Recognition Letters*, vol. 58, pp. 51–60, 2015.
- [37] V. Latora and M. Marchiori, "A measure of centrality based on network efficiency," *New Journal of Physics*, vol. 9, article 188, 2007.
- [38] Y. Mehmood, N. Barbieri, F. Bonchi, and A. Ukkonen, "CSI: Community-level social influence analysis," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics): Preface*, vol. 8189, no. 2, pp. 48–63, 2013.
- [39] C. S. E. Bale, N. J. McCullen, T. J. Foxon, A. M. Rucklidge, and W. F. Gale, "Modeling diffusion of energy innovations on a heterogeneous social network and approaches to integration of real-world data," *Complexity*, vol. 19, no. 6, pp. 83–94, 2014.
- [40] P. De Meo, E. Ferrara, D. Rosaci, and G. M. L. Sarné, "Trust and compactness in social network groups," *IEEE Transactions on Cybernetics*, vol. 45, no. 2, pp. 205–216, 2015.
- [41] H. Liu, Y. Zhang, H. Lin, J. Wu, Z. Wu, and X. Zhang, "How many zombies around you?" in *Proceedings of the 13th IEEE International Conference on Data Mining, ICDM 2013*, pp. 1133–1138, USA, December 2013.
- [42] Z. Chu, S. Gianvecchio, H. Wang, and S. Jajodia, "Detecting automation of Twitter accounts: Are you a human, bot, or cyborg?" *IEEE Transactions on Dependable and Secure Computing*, vol. 9, no. 6, pp. 811–824, 2012.
- [43] Y. Liu, D. Pi, and L. Cui, "Metric Learning Combining With Boosting for User Distance Measure in Multiple Social Networks," *IEEE Access*, vol. 5, pp. 19342–19351, 2017.
- [44] Q. Zhang, J. Wu, Q. Zhang, P. Zhang, G. Long, and C. Zhang, "Dual influence embedded social recommendation," *World Wide Web*, 2017.
- [45] Q. Yan, L. Wu, and L. Zheng, "Social network based microblog user behavior analysis," *Physica A: Statistical Mechanics and its Applications*, vol. 392, no. 7, pp. 1712–1723, 2013.

Research Article

Deep Recurrent Model for Server Load and Performance Prediction in Data Center

Zheng Huang,^{1,2} Jiajun Peng,¹ Huijuan Lian,¹ Jie Guo,¹ and Weidong Qiu¹

¹School of Cyber Security, Shanghai Jiao Tong University, Shanghai, China

²Westone Cryptologic Research Center, Beijing 100070, China

Correspondence should be addressed to Jiajun Peng; pjj@sjtu.edu.cn

Received 31 August 2017; Accepted 2 November 2017; Published 26 November 2017

Academic Editor: Jia Wu

Copyright © 2017 Zheng Huang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Recurrent neural network (RNN) has been widely applied to many sequential tagging tasks such as natural language process (NLP) and time series analysis, and it has been proved that RNN works well in those areas. In this paper, we propose using RNN with long short-term memory (LSTM) units for server load and performance prediction. Classical methods for performance prediction focus on building relation between performance and time domain, which makes a lot of unrealistic hypotheses. Our model is built based on events (user requests), which is the root cause of server performance. We predict the performance of the servers using RNN-LSTM by analyzing the log of servers in data center which contains user's access sequence. Previous work for workload prediction could not generate detailed simulated workload, which is useful in testing the working condition of servers. Our method provides a new way to reproduce user request sequence to solve this problem by using RNN-LSTM. Experiment result shows that our models get a good performance in generating load and predicting performance on the data set which has been logged in online service. We did experiments with nginx web server and mysql database server, and our methods can be easily applied to other servers in data center.

1. Introduction

In the past few decades, the World Wide Web (WWW) [1] has experienced phenomenal growth and server systems have become more and more complex and performance-hungry. With the popularization of B/S structure, data processing is further centralized to the server, which means new challenges to the management of server performance [2]. It is an important part in computer system performance management to predict the server's infrastructure resource performance (like CPU rate and throughput) and workload (user's requests) correctly and effectively, which helps improve quality of the service while minimizing the wastage in resource utilization.

Many applications of big data analytics have been developed to enhance the operation of cloud computing and web server infrastructures in recent years [3]. Previous methods for the performance prediction job basically fall into two categories: one is focusing on building the relation between performance and time, such as neural network (MLP), and linear regression [4], weighted multivariate linear regression

(MVLRL) [5], and recurrent neural network (RNN) [6] even LSTM [7] are used; another one does not consider the sequential effects and predicts the performance by analyzing the workload; for example, Yu et al. [8] use Clustering and Multilayer Perceptron (MLP) to do this task.

Both of these two kinds of method may not explore the essence of the problem. Since the fluctuation of server's performance is caused by user's request sequence, we argue that both user's behavior and its property of the sequence need to be considered when predicting the web server performance. And predicting the performance based on events sequence (user's requests) is our new idea.

The workload in our research is the requests loaded to the server; thus predicting the workload means predicting the requests sequence accessed by the users actually, which has few previous related studies. Previous works in this job focused on predicting the total request situation, for example, the total number of users or the number of requests in a time-window, but did not consider the detail of it, such as the research of Vercauteren et al. [9]. So it is very difficult to reproduce the workload to the server for testing the

performance of servers, which can be very useful for server management.

Like the idea for performance prediction, we argue that users request to servers is the base of the total workload, so the job mentioned here is reproducing the user's access sequence, which can be considered a kind of user characterization.

Recently deep neural networks start to show their great capability in language modeling [10]. And recent research reveals that RNN significantly outperforms popular statistical algorithms [11]. As a special kind of RNN, LSTM neural network [12] is proved to be efficient in modeling sequential data like speech and text [13]. These previous researches inspires us to use LSTM in these prediction tasks, because user's request is a sequential data.

For the purpose of improving the performance of predicting web server performance and workload, we apply RNN-LSTM network with requests-to-vector to this task. Our contributions can be summarized as two points:

- (1) Our work is the first one to apply RNN-LSTM network to predict the performance and workload of Web servers or data center.
- (2) We proposed to investigate the relation between users' requests sequence and web server performance, which previous researches did not pay much attention to.

In a word, our research consists of two models; the workload prediction model in this paper is used to generate an analog workload which is the user's specific request sequence. And the one for performance prediction can predict the performance by analyzing the user's request sequence.

This paper is organized as follows. Section 2 introduces LSTM network and the architecture of our models. Then we introduce our training and application framework in Section 3. Section 4 shows the details and results of our experiments and compares our work with previous researches. Finally, Section 5 is the conclusion of the whole paper.

2. Model

2.1. LSTM Network. As shown in Figure 1, the basic structure of a LSTM unit is composed of a memory cell $c^t \in R^d$ and three essential gates: Input Gate $i^t \in R^d$, Output Gate $o^t \in R^d$, and Forget Gate $f^t \in R^d$.

The formulas for updating the state of each gate and cell in a LSTM unit using the input of x^t , y^{t-1} , and c^{t-1} are defined as follows:

$$\begin{aligned}
 z^t &= g(W_z x^t + R_z y^{t-1} + b_z), \\
 f^t &= \sigma(W_f x^t + R_f y^{t-1} + b_f), \\
 i^t &= \sigma(W_i x^t + R_i y^{t-1} + b_i), \\
 o^t &= \sigma(W_o x^t + R_o y^{t-1} + b_o), \\
 c^t &= i^t \odot z^t + f^t \odot c^{t-1}, \\
 y^t &= o^t \odot h(c^t).
 \end{aligned} \tag{1}$$

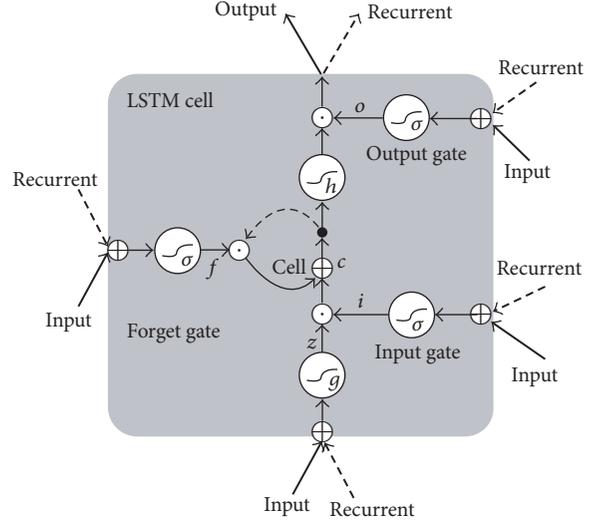


FIGURE 1: Schematic of LSTM unit [14].

Here x^t means the input feature vector at time t . Similarly, y^{t-1} and c^{t-1} are the output vector and cell state at time $t - 1$. And each of them is a d -dimensional value. W in formulas are the weight matrices of the input parts in the gates and cell of LSTM network, and R are the ones of the recurrent parts. b means bias vectors of each formula. As for the \odot mark, it means pointwise multiplication. The $\sigma(x)$, $g(x)$, and $h(x)$ functions are the activation functions of every part in LSTM, which determine the amount of information that can be passed. And we use sigmoid as the activation function of three gates ($\sigma(x)$ in the formulas) and $g(x)$, and we use the rectified linear units (ReLUs) function [15] as the function $h(x)$ in the formulas. ReLU function is a very popular new nonlinear activation function and it is defined as follows:

$$h(x) = \max(0, x). \tag{2}$$

And using ReLU as the activation function can make the network be trained several times faster than using equivalents with saturating neurons like tanh and sigmoid [15].

With this special structure, LSTM network is robust with respect to exploding and vanishing gradient problems [12], so it is able to learn long-term dependencies which RNN cannot perform very well and makes the model be trained without hand-generated features.

Because of the advantages of LSTM, we use LSTM as the basic part of our model to capture the sequential information of requests and then predict the workload and the performance by using the highly abstract features generated by LSTM layers.

2.2. Model for Workload Prediction. One of our preliminary ideas for this model is taking each user's request as an instance and certain length of user's request as a bag containing a number of instances, and the label of the bag is set to be the next request of the final one in the bag, so predicting users requests can be regarded as a problem of multiple-instance

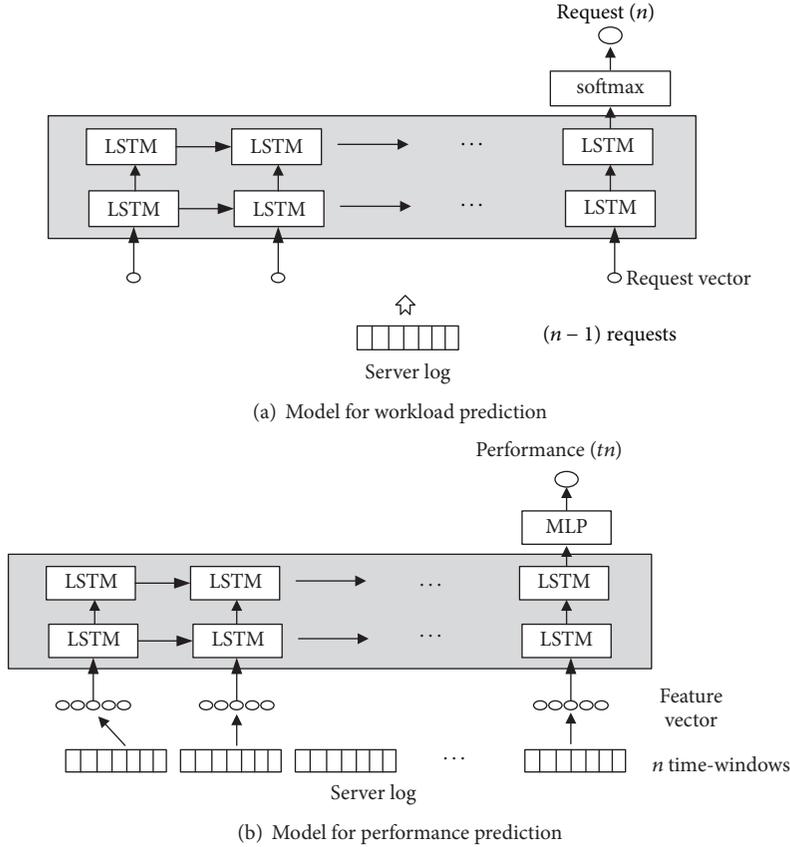


FIGURE 2: The network structure for two models: (a) model for workload prediction and (b) model for performance prediction. Model (a) takes request vectors as the input and output of the network, using the latest $n - 1$ requests to predict the n th request. The input of model (b) is the latest n feature vectors extracted from the log of every time-window, and its output is the value of certain kind of performance in n time-window.

learning (MIL), which has been used widely in drug discovery, text categorization, and graph classification [16]. Previous works like the researches of Wu et al. [17, 18] have shown good classification effectiveness in the area of graph classification. However, the sequential features of the requests are not considered if doing prediction in this way. So we finally selected using LSTM network in this prediction task.

Our model for workload prediction is designed to predict the request sequence of users, which is similar to natural language generation. As shown in Figure 2, the base layer of both our models is two layers of RNN-LSTM, which can capture the features of the user request sequence. And we only use the outputs of the final LSTM unit to do the prediction, which is the many-to-one model of LSTM. This means we use the previous $n - 1$ request to predict the n th request (suppose the length of the sequence is n).

The output layer of the model for workload prediction is obviously a multiclassification task for every user request. So the output of the LSTM layer is designed to be passed into a hidden layer with softmax function, which finally outputs the probability of each request in the $[0, 1]$ interval. Suppose the input vector of the softmax layer is $(\widehat{y}_i^1, \widehat{y}_i^2, \dots, \widehat{y}_i^k)$; we get the

output vector of this layer $(p_i^1, p_i^2, \dots, p_i^k)$ by using following formula:

$$P(\widehat{y}_i^j) = \frac{e^{\widehat{y}_i^j}}{\sum_{k=1}^K e^{\widehat{y}_i^k}}, \quad j \in [1, K]. \quad (3)$$

And the model is trained by minimizing the cross-entropy error between the output of the whole network and the true value. Using cross-entropy as the loss function is the most popular choice in the area of multiclassification task, which also achieves good performance. Suppose the true label is $y_i \in [1, k]$, and the true vector is one-hot, so the loss function is defined as follows:

$$J(\theta) = -\sum_{j=1}^k 1\{y_i = j\} \log P(\widehat{y}_i^j), \quad (4)$$

where θ means the parameters of the network, and the symbol $1\{\cdot\}$ means an indicator that equals one when the condition in the brackets returns true and it equals zero otherwise.

2.3. Model for Performance Prediction. Different from the traditional work of RNN-LSTM in NLP task and the model for workload prediction, of which the output layer is usually

designed as a classifier, the output layer of this model is designed as a liner regression task for every performance. We want to predict the performance like throughput, request delay, and CPU rate, which are not classifications but certain numbers, by analyzing the requests sequence recorded by the server. So the output layer of this model is designed to be a Multilayer Perceptron (MLP), which can fit any continuous function with enough neurons in theory [19].

And ReLU is also used as the activation functions of each fully connected layer here. To prevent the neural networks from overfitting, a dropout [20] layer is connected between two fully connected layers and LSTM layers with the fixed probability $p = 0.5$. This probability means half of the units in the network will be randomly selected and then temporarily removed from the network when training the network. It has been proved by previous work that performance of the networks can be improved dramatically when applying dropout layers at multiple LSTM layers [21].

As shown in Figure 2(b), this model uses the latest requests in n time-windows to predict the performance in the n th time-window, which is a small but important difference between the models for workload, because the performance is affected by not only previous requests but also current operation of users, while the n th request is just strongly related to previous requests.

For the linear regression tasks, L1 norm (absolute differences), L2 norm (squared differences), and smooth L1 [22] are well-known loss functions. The L1 norm is not smooth when the error is close to 0, so it is seldom used. Smooth L1 is a robust L1 loss, and it is less sensitive to outliers than the L2 loss [22]. But all of our data are fed after normalization, and there is little outliers out data set; considering the simplicity of realization, we choose L2 norm as the loss function of this model.

Supposing that the input vector is x_i , \widehat{y}_i is the value predicted by the model, and θ means the network parameters, we use the variance between the real value of the performance and the predicted one $(\widehat{y}_i - y_i)^2$ as the cost of \widehat{y}_i . So the total loss function of the model is defined as follows:

$$J(\theta) = \frac{1}{m} \sum_{i=1}^m (\widehat{y}_i - y_i)^2, \quad (5)$$

which is the mean squared error (MSE) between predicted and true value, and m is the number of sequences in a batch.

3. Training and Application Framework

3.1. Training Framework. Training framework of our models is shown in Figure 2; (a) is the model for workload prediction and (b) is the one for performance prediction.

As shown in Figure 2, log files of servers are the raw data source of our models, so the step for data processing is quite similar, and the main idea of our models is predicting the web server performance and workload by analyzing the log files. Each request has an one-to-one ID in the form of integer which is stored in a dictionary. The dictionary is generated by collecting all the unique request string in the whole data set. Using this ID, the requests can be abstracted

to a one-hot vector with d -dimensional, and we call it request vector. The request vector of the request with ID 1 is $v = (1, 0, 0, \dots)$ and the one with ID 2 is $v = (0, 1, 0, \dots)$ and so on. The feature vectors of requests during a time-window can be generated by adding each request vector, and thus every dimension of which means the number of times user requests the server during a time-window. Take feature vector $v = (p^1, p^2, \dots, p^d)$ as an example; p^i $i \in [1, d]$ in v means that user sends the request whose ID is i altogether p^i times in a time-window.

Actually, this step is not limited to abstract a certain form of user's requests. For example, the request to a database and URL record of a web site also can be done in this way. Its key point is to describe the user's request in a mathematical way, which is using the ID to represent different kinds of user's behavior. And this initialization step is regarded as request-to-vector. After the step of request-to-vector, a long sequence of feature vectors or request vectors can be generated, which is the input of our models.

For the model for workload prediction, previous $n - 1$ request vector should be fed into the network, and the n th request vector is the true label of this request vector sequence. Because of using the many-to-one model of LSTM, the output of the model for workload is the n th request, which is the same as the model for natural language generation. Although the meaning of the label output by the network is different, the principle is the same. So 1 to $n - 1$ request vector is set to be the first sequence, and 2 to n is the second one, and so on, which is similar to the effect of sliding window with the length of $n - 1$. Figure 3(a) shows the process of data set generation for this model.

As for the model for performance prediction, n feature vectors should be fed into the network; the output of the network should be the performance at time t_n . The process of data set generation for this model is shown in Figure 3(b). As the previous feature vectors and the current one both should be put into the network, the process of data set generation has a little difference between the ones shown in Figure 3(a). All of the performance values are normalized by dividing the maximum value in theory before feeding them into the network, which can improve the efficiency of training and the performance of the network [23]. Theoretically, this model is not limited to use the specific kinds of performance, CPU, and memory occupancy rate of the server and other performances also can be trained if data exists.

Finally, sequences of request vector or feature vector are put into the LSTM layer of our models. And the output vector of the LSTM is then passed to the upper layer, which is the softmax layer for the model of workload prediction and the MLP network for the one of performance prediction. And we apply the RMSPROP gradient descent algorithm when training both two networks which is an improved version of SGD algorithm and has better performance with minibatches [24].

3.2. Application Framework. With regard to the use of the model, we propose an application framework, which consists of three kinds of model selection, as shown in Figure 4. We can use the models to predict the workload and performance

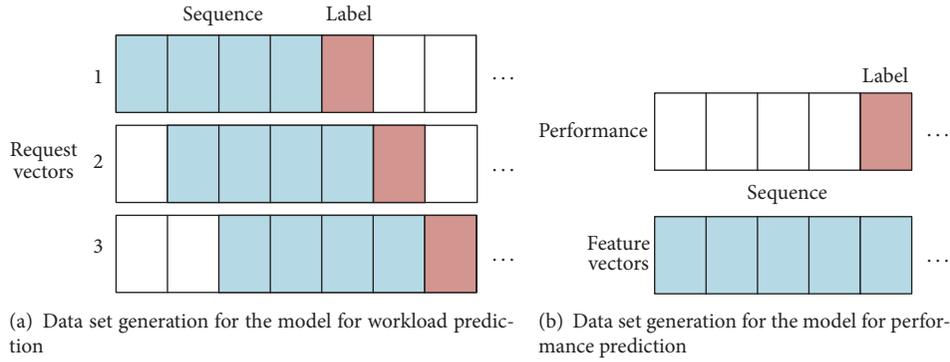


FIGURE 3: Data set generation for two models: (a) model for workload prediction and (b) model for performance prediction. The blue boxes are the input sequence of the model, and the red box means the label of the sequence. (a) Model for workload prediction: input sequence is a certain number of previous request vectors, and the label is the request vector right after the last request vector of the input sequence. (b) Model for performance prediction: input sequence is a certain number of previous feature vectors, and the label is the performance value of the last feature vector of the input sequence.

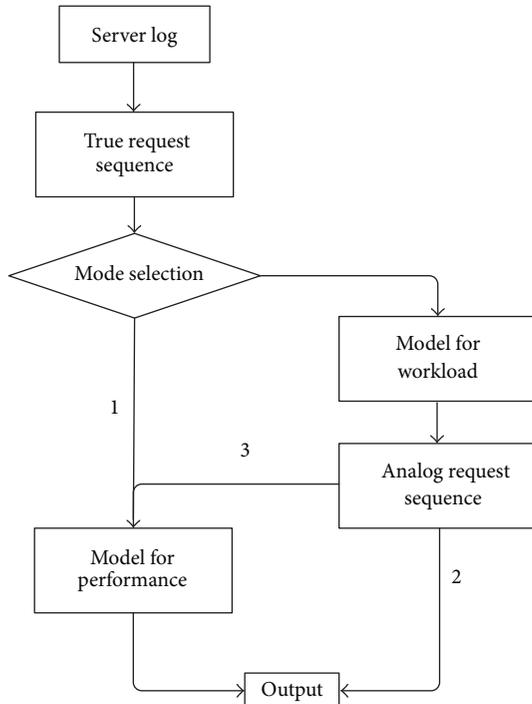


FIGURE 4: Application framework for two models.

of the server by analyzing new log files. The two models can be used separately, which is the basic way to use our models (1, 2 in Figure 4). After training the model for workload prediction, request sequences under different load conditions can be produced by feeding different seeds into the network. The training and application methods of this model are very similar to the usage of RNN-LSTM in the area of natural language generation (NLG). As for the model for performance prediction, it can predict the performance of server under different workload conditions.

And two models can be used in combination; the model for workload prediction can first output analog request sequence and then feed the sequence into the model for

performance prediction (3 in Figure 4). By using the model for workload prediction, request sequence with the original load features can be generated. This model of usage can meet the needs such as when the data of new log is too little, and the performance under a long-term workload is required. With these 3 kinds of options to apply these two models, the models can be used more flexibly and adapted to more situations.

On the other hand, our models are not limited by the network architecture theoretically. Our models can be deployed on the load balance node to predict the operation situation of the whole network or just deployed on the service nodes like data center node or calculation node to predict the operation situation of certain node.

In a word, our models can meet many different prediction needs regardless of the network architecture, which could be helpful in the management of data center.

4. Experiment

4.1. Setup. We keep the length of LSTM network to 15 (15 seconds), because we assume that a request cannot affect the web server performance and workload after 15 seconds in general. So the requests in each 15 seconds are organized into one sequence. Our model is trained and tested on the GPU: NVIDIA GeForce GTX 1080Ti, and the model was developed on the framework of theano and lasagne with CUDA to accelerate calculation. It took about 6 to 7 hours to finish the training of the model on the GPU. As a comparison, it will take more time to complete this job on an ordinary CPU.

For the model for workload prediction, it is very hard to measure the degree of similarity between the analog load and the real load. In our research, we just do the preliminary measure, which uses the difference between real and simulated proportion of each ID of request to measure the efficiency of this model. So cosine similarity is used to do this job, and the definition is shown follows:

$$\text{similarity} = \cos(\theta) = \frac{\sum_{i=1}^n A_i B_i}{\sqrt{\sum_{i=1}^n A_i^2} \sqrt{\sum_{i=1}^n B_i^2}}, \quad (6)$$

where A and B are the proportion vectors of the true one and generated one, and A_i and B_i are components of vectors A and B , respectively, which means the proportion of the request with ID i . So the value of the similarity is $[0, 1]$, and the closer the number is to 1, the more similar the two vectors are.

As for the one for performance prediction, we use mean squared error (MSE), which is also the loss function of the network to measure the efficiency of the model.

4.2. Experiment on Data Set A

4.2.1. Data Set. Data set A for evaluating the performance of our models contains the log files of 191 web servers nodes in one day. The web servers are set up using nginx and have been deployed in production environment, which means the records in log files are real. The log files of nginx can record the request in the form of URL. Besides the URL sequences, we can also get status code, the delay of each request, and the number of bytes of data transmission for each request from the log files. We can obtain the server's error rate, throughput, and average request delay, which is the three performances in this experiment. Table 2 shows the performance we tested in this experiment and their description. We regard the maximum throughput of the network card as the maximum of throughput, which is 12.5 Mb/S. 100 ms is the maximum of request delay. As for error rate, it is in $[0, 1]$ interval originally.

Because of the limit of GPU memory and the large number of log files, we choose to use log files of several random nodes which has the biggest log files to train the model.

4.2.2. Data Processing. First of all, we filter some requests which are not user's main operation, such as the requests for a picture or a json data. Then we get 57453 valid requests, which contain 2049 different URLs. For the model for workload prediction, all of the requests were set as training set. After requests-to-vector for URL requests and calculation of the performance mentioned in Section 3.1, a one-to-one relationship with URL requests and performance can be established. Finally, the data sets in the form of sequence according to the chronological order of URL requests are generated. We split the whole data sets into training set and test set, in accordance with the ratio of four to one. And one-fifth of the training set are set as validation set.

4.2.3. Result. In the experiment for performance prediction, three performances including request error rate, throughput, and request delay are simulated. The features of the network which has the best result on the validation set are saved, and then we perform a final test to see the performance of our network by using the test set.

Figure 7 shows a declining trend of the error in training, which means our model converges quite well. Because the value of objective function $(\hat{y}_i - y_i)^2$ is very small in this experiment, we draw the picture using the data of $\log(\hat{y}_i - y_i)^2$ which can show the trend more clearly. After training about

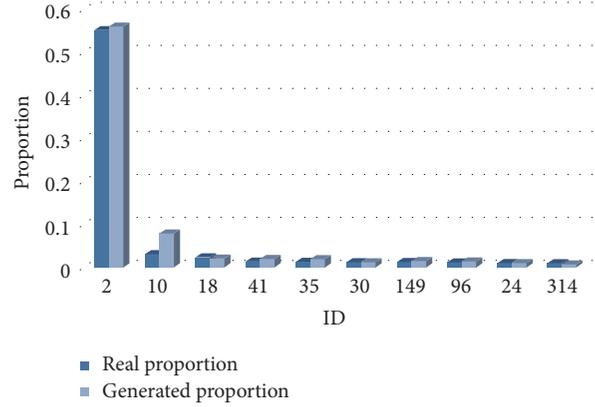


FIGURE 5: Proportion of top 10 ID on data set A.

200 epochs, the model gets the best result in training set and validation set.

Table 3 is the data of the final result about the objective function $(\hat{y}_i - y_i)^2$ on the test set, compared with the results on the training set. The data of average $(\hat{y}_i - y_i)^2$ in Table 3 shows that this model performs well in the task of predicting web server performance, which is a new area of using LSTM. On the other hand, the results of the two data sets (validation set and test set) are similar, so we think the model has a certain generalization ability.

In the experiment for workload prediction, the model generated 43077 URL requests with the same number of different request IDs. Because the number of different ID is too much, and most of the requests just appear so little times, we just compare the proportion vectors of top 10 frequent requests. Table 1 and Figure 5 show the result of this experiment. As shown in the table, the proportion of most kinds of request is less than 1%, and more than half of the requests are the request with ID: 2.

The cosine similarity is 0.996607614, which means the workload generated is very similar to the real workload and the model can capture the features of request sequence and regenerate them.

4.3. Experiment on Data Set B

4.3.1. Data Set. Because the requests in data set A concentrated in several requests, and the number of all requests is too much, we test our models on another data set, which we call it data set B. Different from A, data set B contains the log files of a database which also has been deployed in production environment. The requests of users are much more simple, and it contains only 4 types of requests. On the other hand, the performance data in this data set is recorded by other tools, so the logs of the server only provide the user request sequences and performance data is provided by the tools. Table 4 shows the performance we tested in this experiment and its description. We regard the maximum value in the data set as the maximum of average delay and

TABLE 1: Proportion of top 10 ID on data set A.

ID	Proportion									
	2	10	18	41	35	30	149	96	24	314
Real proportion	0.5497	0.0305	0.0226	0.0125	0.0124	0.0123	0.0123	0.0117	0.0111	0.0106
Generated proportion	0.5603	0.0789	0.0188	0.0193	0.0206	0.0128	0.0153	0.0141	0.0110	0.0083

TABLE 2: Performance and description on data set A.

Performance	Description
Error rate	The percentage of requests errors (like 404, 403,...)
Throughput	The total amount of data the server transmits
Request delay	The average delay in processing user requests

TABLE 3: Result of model for different performance on data set A.

Performance	Training set	Validation set	Test set
Error rate	$1.61 * 10^{-6}$	$1.71 * 10^{-06}$	$2.89 * 10^{-5}$
Throughput	$1.59 * 10^{-15}$	$1.60 * 10^{-15}$	$1.37 * 10^{-15}$
Request delay	$5.65 * 10^{-12}$	$6.33 * 10^{-12}$	$6.48 * 10^{-12}$

TABLE 4: Performance and description on data set B.

Performance	Description
CPU rate	The occupancy rate of the server's CPU
Average delay	Average delay of the requests
QPS	Query per second

TABLE 5: Result of model for different performance on data set B.

Performance	Training set	Test set
CPU rate	$1.002635 * 10^{-4}$	$8.360695 * 10^{-4}$
Average delay	$8.25704107 * 10^{-5}$	$2.9352345 * 10^{-3}$
QPS	$1.039598 * 10^{-4}$	$1.413895 * 10^{-4}$

QPS to do normalization. As for CPU rate, it is in $[0, 1]$ interval originally.

4.3.2. Result. In the experiment for performance prediction, three performances including CPU rate, average delay, and QPS are simulated. Table 5 is the data of the final result about the loss function $(\hat{y}_i - y_i)^2$, and the error of the predicted value is acceptably small.

In the experiment for workload prediction, the model generated 2000 request sequences with the length of 15 finally. The statistical result is shown in Table 6 and Figure 6, and the cosine similarity is 0.9985807, so the angle between two vectors is about 3° , which means that the workload is quite similar at the level of proportion.

The result on this data set shows that our model also performs well when the requests are in the form of the access to the database, which proves our theory that the form of the records or the requests does not affect the results of our model. And our model has the ability to adapt to a variety of situations.

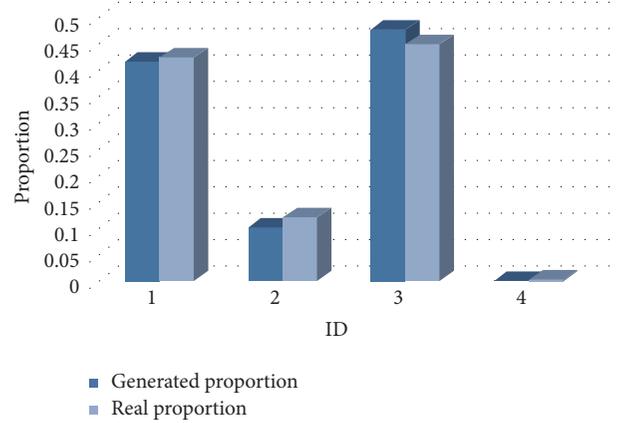


FIGURE 6: Proportion of each ID on data set B.

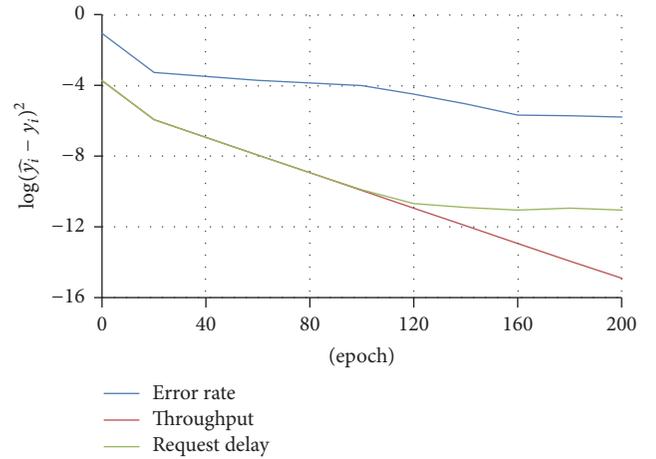


FIGURE 7: Illustration of declining trend of the error in training.

TABLE 6: Proportion of each ID on data set B.

Type (ID)	Generated proportions	Real proportions
1	0.41843333	0.4262904
2	0.10083333	0.1205442
3	0.48056667	0.4510679
4	0.00016667	0.0020976

In conclusion, the recurrent network can predict the web server performance and workload by analyzing the request sequences and turn out to have good performance both in accuracy and in generalization ability.

4.4. Related Works. Both of our models are built based on event (requests of users), which is a totally new method for

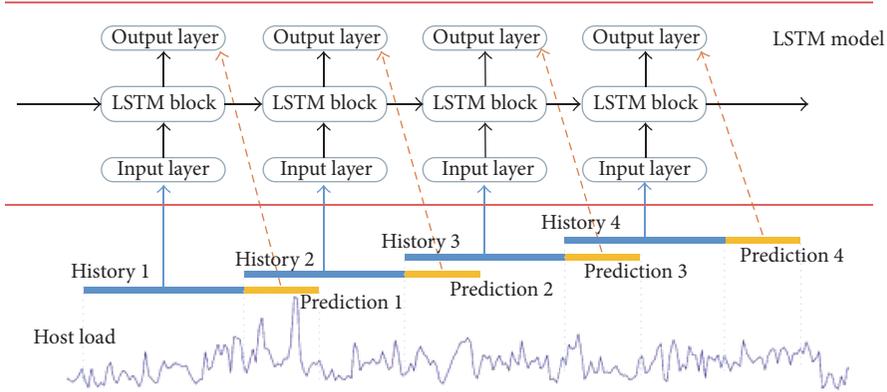


FIGURE 8: Model of Song et al. [7].

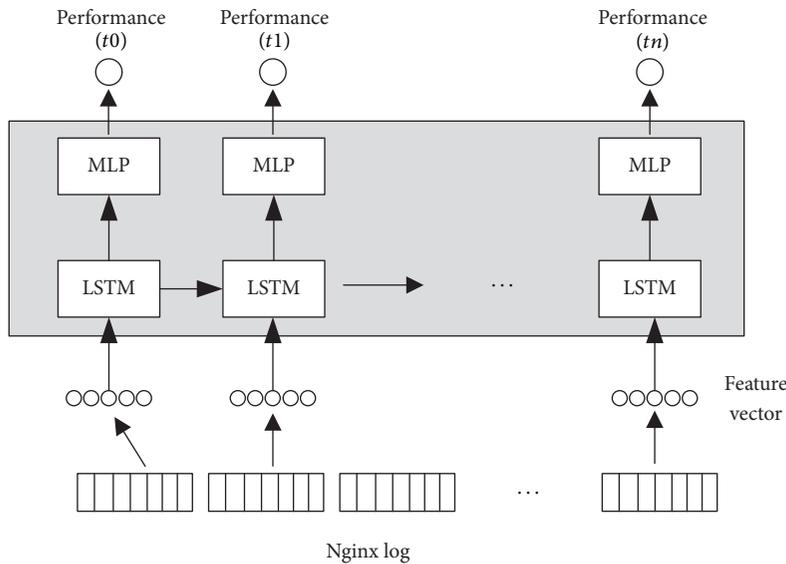


FIGURE 9: Model of our previous work.

TABLE 7: Result of model of Song et al. on data set B.

Performance	Training set	Test set
CPU rate	3.6827354×10^{-3}	4.0699269×10^{-3}
Average delay	6.2893999×10^{-4}	2.7876459×10^{-3}
QPS	1.3815304×10^{-4}	2.3197048×10^{-3}

the prediction task of the server or data center. So our model for performance prediction is very different from previous works in this prediction task. Take the work of Song et al. [7] as an example, which also uses LSTM in their work.

As shown in Figure 8, their work uses the previous performance to predict the future one, which means fitting the curve of performance with time. And it is also the key idea of previous work, but the fitting method of each is not the same.

As a comparative experiment, we did an experiment on data set B using the model of Song et al. [7]. The result is shown in Table 7, and the mean squared error (MSE) of each

performance is larger than ours, which means our model achieves a better result.

As for the model for workload prediction, there is so little previous research, so we did not do the comparative experiment.

On the other hand, the research in this paper is an extension of our previous work [25], as shown in Figure 9. We change the model for performance prediction from many-to-many to many-to-one version of LSTM, and we propose the model for load prediction and the application framework of two models to improve our research.

In a word, our work is completely a new method to do this prediction job for server and data center.

5. Conclusion

In this paper, we propose to use RNN-LSTM to predict web server performance and workload. Model for performance prediction is composed of RNN-LSTM and Multilayer Perceptron (MLP), and the one for workload prediction consists

of RNN-LSTM and softmax layer. Doing the research based on events is a new way in this prediction area. The models can extract features automatically during the learning process without any prior knowledge or hand-generated features for segmentation. Experiments conducted on real data sets show that our models can achieve a good performance and generalization on predicting the performance of different kinds of servers. And the result also shows that the load generated by our model is very similar to the real one, which can be applied to test data center and other kinds of servers. Our results suggest that RNN-LSTM performs well on sequential tagging tasks; moreover RNN-LSTM with requests-to-vector is a new effective method to predict server performance and workload which is worth further exploration. Most servers in data center has log system. As long as the log file recording the operation of the users is provided, our method can be used to generate load for the server and predict server performance under different load conditions. This can save a lot operation and maintenance work in data center.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work is supported by the National Key Research and Development Program of China under Grants 2017YFB0802704 and 2017YFB0802202 and Program of Shanghai Technology Research Leader under Grant 16XD1424400.

References

- [1] T. Berners-Lee, R. Cailliau, J.-F. Groff, and B. Pollermann, "World-wide web: The information universe," *Internet Research*, vol. 20, no. 4, pp. 461–471, 2010.
- [2] J.-F. Tu and R.-F. Guo, "The application research of mixed program structure based on client-server, browser-server and web service," in *Proceedings of the 2011 International Conference on Business Management and Electronic Information, BMEI 2011*, pp. 193–195, May 2011.
- [3] R. Buyya, K. Ramamohanarao, C. Leckie, R. N. Calheiros, A. V. Dastjerdi, and S. Versteeg, "Big data analytics-enhanced cloud computing: Challenges, architectural elements, and future directions," in *Proceedings of the 21st IEEE International Conference on Parallel and Distributed Systems, ICPADS 2015*, pp. 75–84, December 2015.
- [4] S. Islam, J. Keung, K. Lee, and A. Liu, "Empirical prediction models for adaptive resource provisioning in the cloud," *Future Generation Computer Systems*, vol. 28, no. 1, pp. 155–162, 2012.
- [5] I. Davis, H. Hemmati, R. C. Holt, M. W. Godfrey, D. Neuse, and S. Mankovskii, "Storm prediction in a cloud," in *Proceedings of the 2013 5th International Workshop on Principles of Engineering Service-Oriented Systems, PESOS 2013*, pp. 37–40, May 2013.
- [6] B. Luo and S.-w. Ye, "Server performance prediction using recurrent neural network," *Computer Engineering and Design*, vol. 8, p. 57, 2005.
- [7] B. Song, Y. Yu, Y. Zhou, Z. Wang, and S. Du, "Host load prediction with long short-term memory in cloud computing," *The Journal of Supercomputing*, pp. 1–15, 2017.
- [8] Y. Yu, V. Jindal, I. Yen, and F. Bastani, "Integrating Clustering and Learning for Improved Workload Prediction in the Cloud," in *Proceedings of the 2016 IEEE 9th International Conference on Cloud Computing (CLOUD)*, pp. 876–879, San Francisco, Calif, USA, June 2016.
- [9] T. Vercauteren, P. Aggarwal, X. Wang, and T.-H. Li, "Hierarchical forecasting of web server workload using sequential Monte Carlo training," *IEEE Transactions on Signal Processing*, vol. 55, no. 4, pp. 1286–1297, 2007.
- [10] M. Auli, M. Galley, C. Quirk, and G. Zweig, "Joint language and translation modeling with recurrent neural networks," in *EMNLP*, vol. 3, 8 edition, 2013.
- [11] M. Osawa, H. Yamakawa, and M. Imai, in *Proceedings of the Neural formation processg-23rd ternational conference, iconip 2016*, Springer Verlag, 2016.
- [12] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.
- [13] M. Sundermeyer, H. Ney, and R. Schluter, "From feedforward to recurrent LSTM neural networks for language modeling," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 23, no. 3, pp. 517–529, 2015.
- [14] X. Ma and E. Hovy, "End-to-end Sequence Labeling via Bi-directional LSTM-CNNs-CRF," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1064–1074, Berlin, Germany, August 2016.
- [15] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proceedings of the 26th Annual Conference on Neural Information Processing Systems (NIPS '12)*, pp. 1097–1105, December 2012.
- [16] J. Wu, S. Pan, X. Zhu, C. Zhang, and X. Wu, "Positive and Unlabeled Multi-Graph Learning," *IEEE Transactions on Cybernetics*, vol. 47, no. 4, pp. 818–829, 2016.
- [17] J. Wu, S. Pan, X. Zhu, C. Zhang, and P. S. Yu, "Multiple Structure-View Learning for Graph Classification," *IEEE Transactions on Neural Networks and Learning Systems*, vol. PP, no. 99, pp. 1–16, 2017.
- [18] J. Wu, S. Pan, X. Zhu, C. Zhang, and X. Wu, "Multi-instance Learning with Discriminative Bag Mapping," *IEEE Transactions on Knowledge and Data Engineering*, vol. PP, no. 99, pp. 1–14, 2017.
- [19] K. Hornik, M. Stinchcombe, and H. White, "Multilayer feedforward networks are universal approximators," *Neural Networks*, vol. 2, no. 5, pp. 359–366, 1989.
- [20] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.
- [21] V. Pham, T. Bluche, C. Kermorvant, and J. Louradour, "Dropout Improves Recurrent Neural Networks for Handwriting Recognition," in *Proceedings of the 14th International Conference on Frontiers in Handwriting Recognition, ICFHR 2014*, pp. 285–290, September 2014.
- [22] R. Girshick, "Fast R-CNN," in *Proceedings of the 15th IEEE International Conference on Computer Vision (ICCV '15)*, pp. 1440–1448, December 2015.
- [23] S. Ioffe and C. Szegedy, "Batch normalization: accelerating deep network training by reducing internal covariate shift," *Learning*, vol. 3, 2015.

- [24] G. Hinton, N. Srivastava, and K. Swersky, *Neural networks for machine learning lecture 6a overview of mini-batch gradient descent*, Coursera Lecture slides, 2012.
- [25] J. Peng, Z. Huang, and J. Cheng, "A Deep Recurrent Network for Web Server Performance Prediction," in *Proceedings of the 2017 IEEE Second International Conference on Data Science in Cyberspace (DSC)*, pp. 500–504, Shenzhen, China, June 2017.

Research Article

Deep Hierarchical Representation from Classifying Logo-405

Sujuan Hou,^{1,2} Jianwei Lin,¹ Shangbo Zhou,³ Maoling Qin,¹
Weikuan Jia,¹ and Yuanjie Zheng^{1,2,4}

¹School of Information Science and Engineering, Shandong Normal University, Jinan 250014, China

²Institute of Life Sciences at Shandong Normal University, Jinan 250014, China

³School of Computer Science, Chongqing University, Chongqing 400030, China

⁴Key Laboratory of Intelligent Information Processing at Shandong Normal University, Jinan 250014, China

Correspondence should be addressed to Yuanjie Zheng; zhengyuanjie@gmail.com

Received 30 June 2017; Accepted 29 August 2017; Published 10 October 2017

Academic Editor: Jia Wu

Copyright © 2017 Sujuan Hou et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

We introduce a logo classification mechanism which combines a series of deep representations obtained by fine-tuning convolutional neural network (CNN) architectures and traditional pattern recognition algorithms. In order to evaluate the proposed mechanism, we build a middle-scale logo dataset (named Logo-405) and treat it as a benchmark for logo related research. Our experiments are carried out on both the Logo-405 dataset and the publicly available FlickrLogos-32 dataset. The experimental results demonstrate that the proposed mechanism outperforms two popular ways used for logo classification, including the strategies that integrate hand-crafted features and traditional pattern recognition algorithms and the models which employ deep CNNs.

1. Introduction

A logo is a symbolic representation of any enterprise or organization or institution, which symbolizes the product or service of their respective work. Logos can be composed of a glyph, a textual message, an icon, or an image, depicted in various colors and styles. Detection and recognition of logos has always been important in a wide range of applications, such as product or brand identification, copyright infringement detection, contextual advertise placement, vehicle logo for intelligent traffic-control system [1], and brand-related statistics from social media streams [2]. At present, with the rapid development of multimedia information technology, the amount of logo data on the Internet continues to grow. Because of the surge in the amount of logos, designing effective management tools and systems is becoming imperative. This paper focuses on developing a fundamental tool for organizing logos by classifying them. Categorizing makes browsing and searching for logos more efficient and facilitates the development of related applications. For instance, in order to ensure the originality and uniqueness, when creating a logo for a new product or organization, it would be useful to be

able to search through similar products or organizations to avoid trade infringement or duplication.

According to Bengio et al. [3], learning representation of the data makes it easier to extract useful information when building classifier. Hence, the success of classification algorithm largely depends on data representation because different representations can entangle and hide more or less the different explanatory factors of variation behind the data. At present, the study of representation for classification has attracted considerable attentions and it has had extensive applications, such as graph representation [4, 5] for classification, advertising video representation [6] for classification, logo classification [7–9], and other classifications employing various technologies, for example, bag mapping for the multi-instance learning [10]. Regarding the logo classification, Neumann et al. [7] classify the logos of University of Maryland logo database by combining local and global shape features. Sun and Chen [8] design a logo classification system to differentiate the logo images captured through mobile phone cameras with a limited set of images. Kumar et al. [11] propose a logo classification system based on the appearance of logo

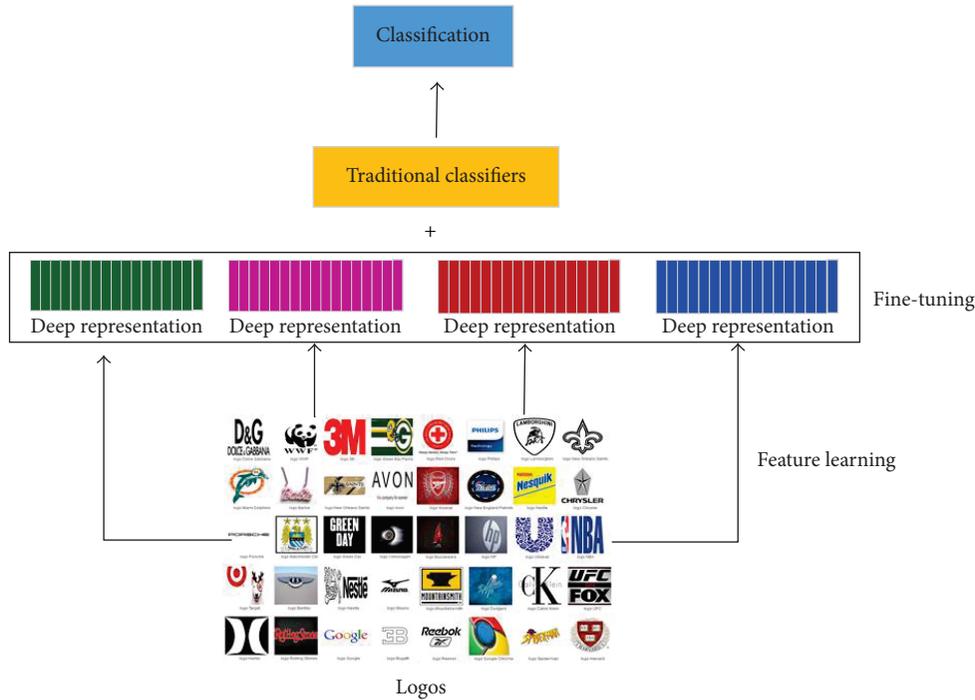


FIGURE 1: Overview of the proposed scheme.

images, which makes use of global characteristics of logo images for classification, like color, texture, and shape.

However, the success of most of existing work on classification, including logo classification, which adopts traditional pattern recognition algorithms primarily depend on the chosen class of features. These chosen features usually tend to be hand-crafted. A recent advance has been the use of deep neural networks to automate visual feature extraction in various domains. In particular, methods that use the convolutional neural network (CNN) model have achieved state-of-the-art results in computer vision tasks. However, as we know, training deep neural network is difficult due to its tendency to have many local optima. Nair and Hinton [12] address this problem by pretraining the deep model, which is called “greedy layerwise training.” Recently, Bianco et al. [13] present a recognition pipeline specifically for logo using deep learning, which is composed of a logo region proposal followed by a CNN.

Considering that the methods adopting a CNN model have shown good performance in image style classification as well when pretrained modes are sufficiently fine-tuned, in this paper, we propose a mechanism that makes full use of both the advantages of fine-tuning CNN models and traditional pattern recognition algorithms for logo classification task. Specifically, we firstly fine-tune several of important deep learning models to obtain the logo representations and then combine the learned logo representations into traditional classification algorithms. Due to the limited amount of training data available for logo task, the deep models work on

networks pertained on other large-scale image datasets. The contribution of this work is twofold:

- (1) We build a publicly available logo dataset (named Logo-405), which can be shared in the research of logos.
- (2) We present a logo classification mechanism that combines both the advantages of deep hierarchical convolutional neural networks and traditional pattern recognition algorithms.

The remainder of this paper is organized as follows: Section 2 provides a description of the proposed mechanism; the experimental results and analysis are presented in Section 3; and Section 4 concludes this paper.

2. Proposed Approach

2.1. Overview. Figure 1 illustrates an overall workflow of the proposed scheme. It contains two stages; they are (1) feature learning phase, in which several deep representations for each logo are obtained by fine-tuning four popular deep convolutional network architectures and (2) classification phase, where the logo classification task is carried out by combining both the learned deep representations and traditional classification algorithms.

The proposed scheme combines both advantages of convolutional neural network in feature learning and traditional classification algorithm. During which four popular deep convolutional neural network architectures are firstly fine-tuned on our logo dataset (i.e., Logo-405) and one publicly

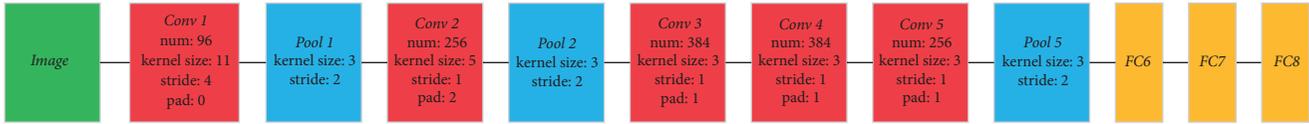


FIGURE 2: Network architecture of AlexNet [14].

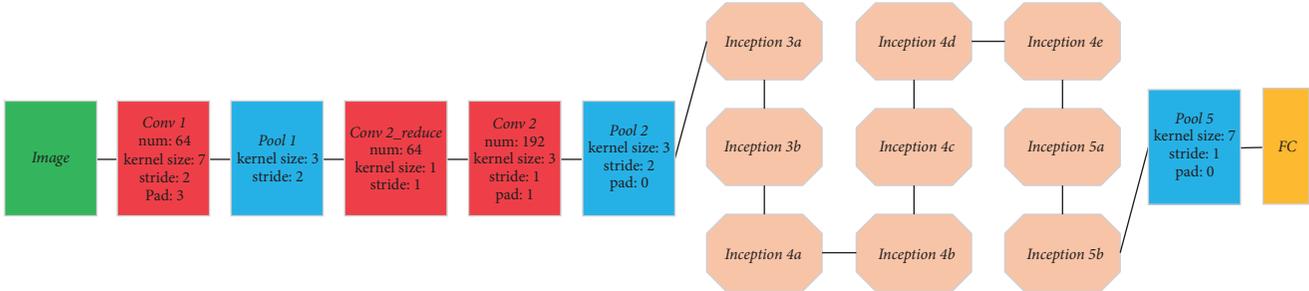


FIGURE 3: Fine-tuned network architecture of GoogleNet [15].

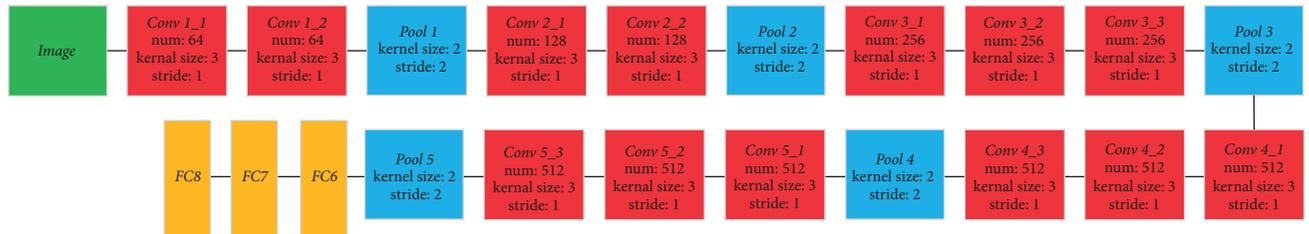


FIGURE 4: Fine-tuned network architecture of VGGNet [16].

available FlickrLogos-32 dataset, respectively. After that, four different deep representations are obtained for each logo image. Then, these learned deep representations are used to differentiate logo categories by training traditional classification models.

2.2. Transfer Learning by Fine-Tuning Deep CNNs. Convolutional neural networks (CNNs) [17] have been proven to be able to achieve great success in computer vision tasks, especially visual feature extraction.

Deep architectures of CNNs, called “deep convolutional neural networks (DCNNs),” have made much success in the ImageNet Large-Scale Visual Recognition Challenge (ILSVRC). There are several popular models for deep convolutional network architectures, including AlexNet [14], GoogleNet [15], VGGNet [16], and ResNet [18].

The early layers of these DCNNs are trained with a large dataset (ImageNet [19] is the common) to extract generic features. In this work, we use methods that fine-tune a pretrained model limited by the scale of logo dataset. Specifically, we use the AlexNet, GoogleNet, VGGNet, and ResNet implementation, respectively, trained on the ImageNet dataset as the pretrained models. In our transfer learning approach, as our dataset is relatively small (32,218 images) compared to ImageNet, we suppose fine-tuning the last layer of the deep models instead of the earlier layers would improve performance. To be detailed, we fine-tune the second-to-last layer of the deep models and initialize the last

full connection layer to 405 outputs, corresponding to 405 categories of logo, to avoid training the model from scratch for classification.

Figures 2–5 show details of four fine-tuned network architectures.

3. Experiments

3.1. Datasets. To evaluate the performance of the proposed mechanism, two datasets are adopted in the experiments, including Logo-405 and FlickrLogos-32 [20].

Logo-405 is a logo dataset crawled from Internet. It contains 405 categories of logos and 32218 logo images are included in total. To the best of our knowledge, Logo-405 is the largest logo dataset up to now. Figure 6 illustrates the logo images that are selected, one from each category.

Another benchmark dataset, named FlickrLogos-32, is a publicly available collection of logo photos. It contains 32 different logo brands by downloading them from Flickr. For each class, the dataset offers 10 training data images, 30 validation images, and 30 test images. An example of logo image of each class from FlickrLogos-32 dataset is illustrated in Figure 7.

3.2. Baseline Representation Methods. To validate the effectiveness of the proposed classification scheme, we compared the proposed method with other several baselines, including global-feature-based approach, local-feature-based method,

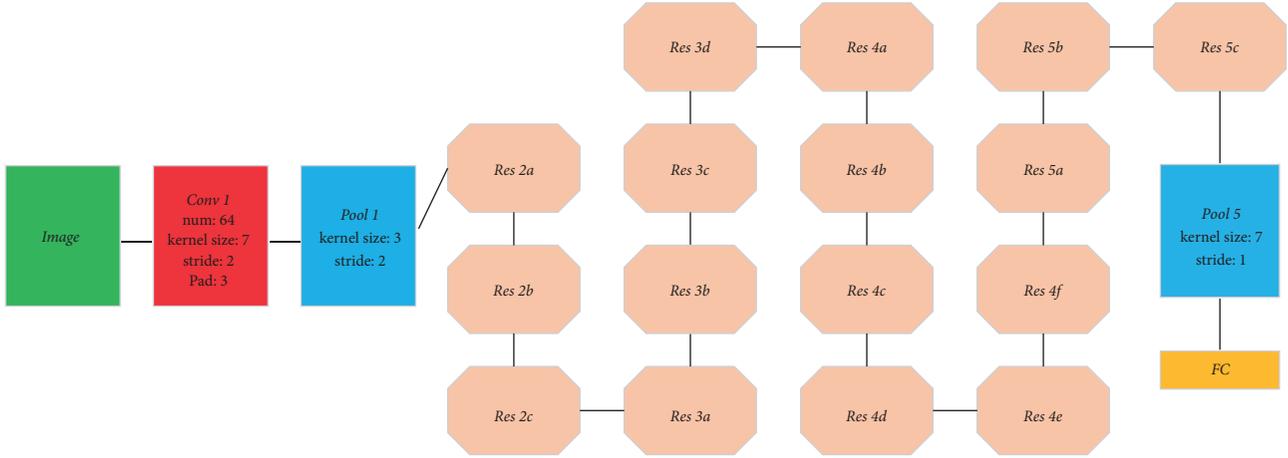


FIGURE 5: Fine-tuned network architecture of ResNet [18].



FIGURE 6: Logo image of each class from Logo-405.

and the models by fine-tuning deep CNNs. They are as follows:

- (i) *Global-feature-based representation (GFBR)*: since the HSV (hue-saturation-value) space conforms to the more similarity of human perception, we adopted the quantized HSV histogram.
- (ii) *Local-feature-based representation (LFBR)*: SIFT [21], as a typical local visual descriptor, has been proved to be able to capture sufficiently discriminative local elements with some invariant properties to geometric or photometric transformations and is robust to occlusion. We first perform hierarchical k -means

in the training set to form a 10000-centered SIFT visual vocabulary and then adopt BOW (Bag-of-Word) technique to build the logo representation. The SIFT feature description was built followed by [1].

- (iii) *Fine-tuning AlexNet representation (FTAN)*: it is a deep representation of logo image by fine-tuning AlexNet architecture. For Logo-405 dataset, the training was performed using stochastic gradient decent with image batch size of 32 images and the learning rate was reduced by hand after 54.42 K iterations from an initial setting of $1e^{-3}$, while, with respect to FlickrLogos-32, the training was also performed using stochastic gradient decent with image batch size of

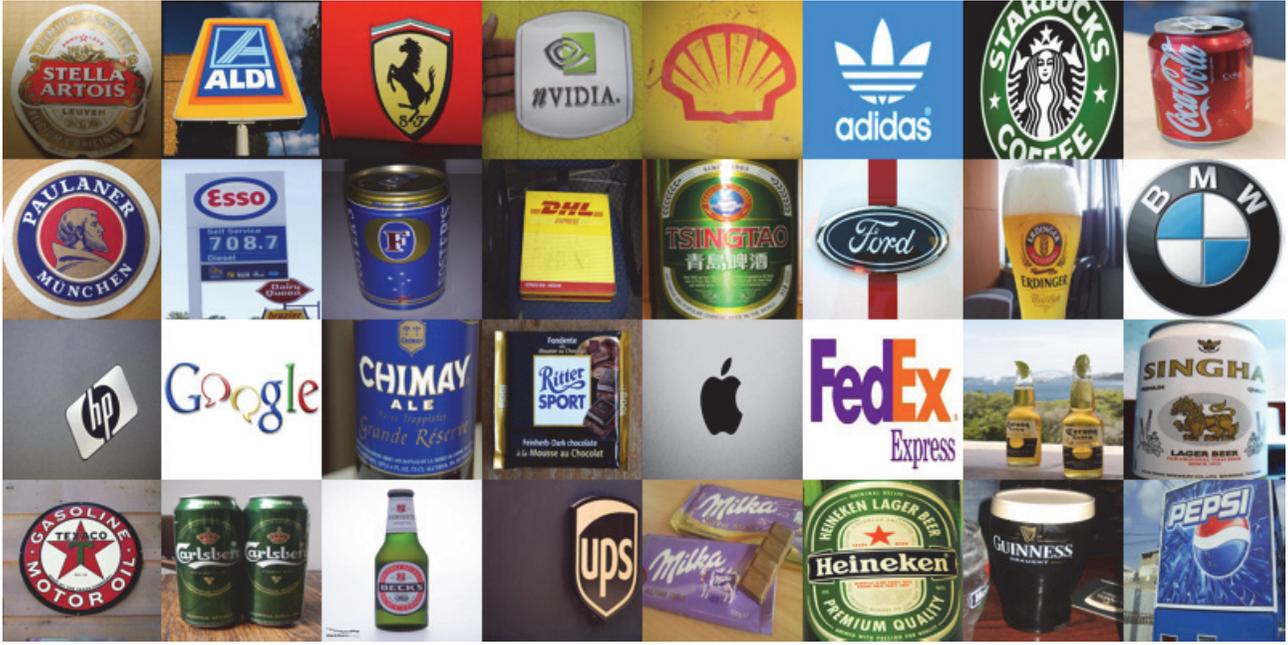


FIGURE 7: Logo images of each class from FlickrLogos-32.

32 images and the learning rate was reduced by hand after 1.89 K iterations from an initial setting of $1e^{-3}$.

- (iv) *Fine-tuning GoogleNet representation (FTGN)*: it is a deep representation of logo image by fine-tuning GoogleNet architecture. For Logo-405 dataset, the training was performed using stochastic gradient decent with image batch size of 32 images and the learning rate was reduced by hand after 54.42 K iterations from an initial setting of $1e^{-3}$, while, with respect to FlickrLogos-32, the training was also performed using stochastic gradient decent with image batch size of 32 images and the learning rate was reduced by hand after 1.89 K iterations from an initial setting of $1e^{-3}$.
- (v) *Fine-tuning VGG representation (FTVGG)*: it is a deep representation of logo image by fine-tuning VGG architecture. For Logo-405 dataset, the training was performed using stochastic gradient decent with image batch size of 32 images and the learning rate was reduced by hand after 54.42 K iterations from an initial setting of $1e^{-3}$, while, with respect to FlickrLogos-32, the training was also performed using stochastic gradient decent with image batch size of 32 images and the learning rate was reduced by hand after 1.89 K iterations from an initial setting of $1e^{-3}$.
- (vi) *Fine-tuning ResNet representation (FTRN)*: it is a deep representation of logo image by fine-tuning ResNet architecture. For Logo-405 dataset, the training was performed using stochastic gradient decent with image batch size of 8 images and the learning rate was reduced by hand after 217.59 K iterations from an initial setting of $1e^{-3}$, while, with respect to

FlickrLogos-32, the training was also performed using stochastic gradient decent with image batch size of 8 images and the learning rate was reduced by hand after 7.56 K iterations from an initial setting of $1e^{-3}$.

- (vii) *Deep architecture in [13]*: it is a CNN network architecture specifically trained on FlickrLogos-32 for logo classification.

3.3. Experiment Setup. For GFBR, considering that color is one of the most dominant and distinguishable global visual feature when describing an image, we define it in terms of a histogram in the quantized hue-saturation-value (HSV) color space with 256 components (H = 16 bins, S = 4 bins, and V = 4 bins).

With regard to LFBR, as previously described, the SIFT was extracted from each logo image and treated as local features. When carrying out LFBR in our task, all the SIFT features were quantized into 10,000 visual words using hierarchical k -means clustering technique.

With respect to the deep representations, the hyper parameter setting used in deep architecture is elaborated as in Section 3.2. Other parameters are adopted as their propositional setting value in [14–16, 18].

For classification algorithms, many classical models and their variants have been proposed, such as SVM [22, 23] and ensemble classifier [23]. In our experiments, 10-fold cross validation was conducted by adopting three classical classifiers, including k NN, random forest, and SVM.

Based on the experimental results of 10-fold cross validation, the performance of each strategy was measured by evaluating the mean average accuracy (MAA) and standard deviation (SD).

TABLE 1: Classification results by adopting fine-tuning deep CNN architectures.

(%)	FTAN	FTGN	FTVGG	FTRN
MAA	77.9084	82.3639	84.9192	84.8881

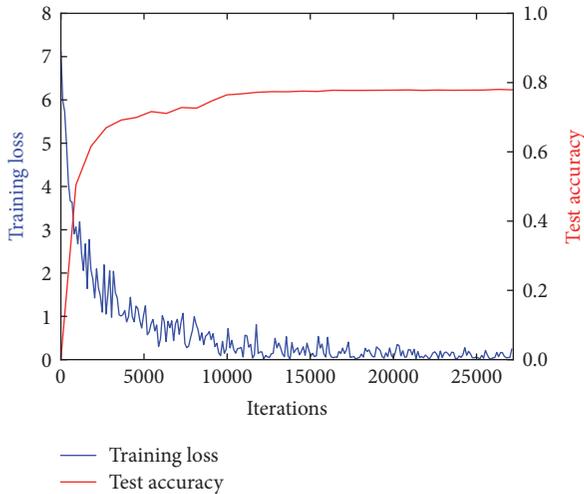


FIGURE 8: The training process of FTAN.

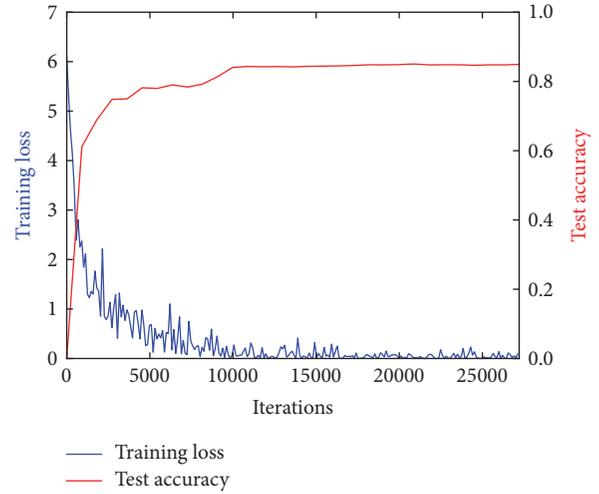


FIGURE 10: The training process of FTVGG.

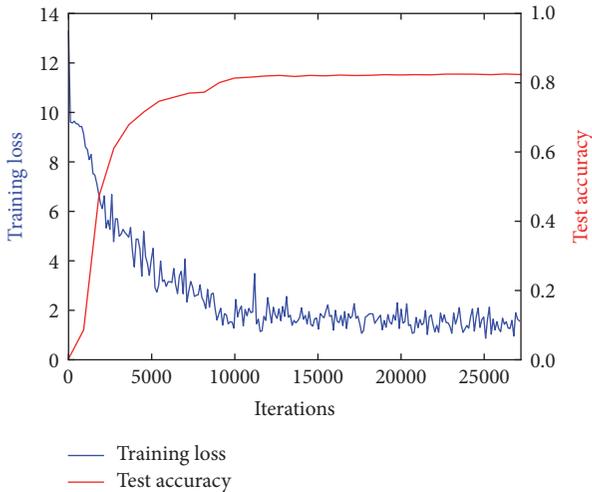


FIGURE 9: The training process of FTGN.

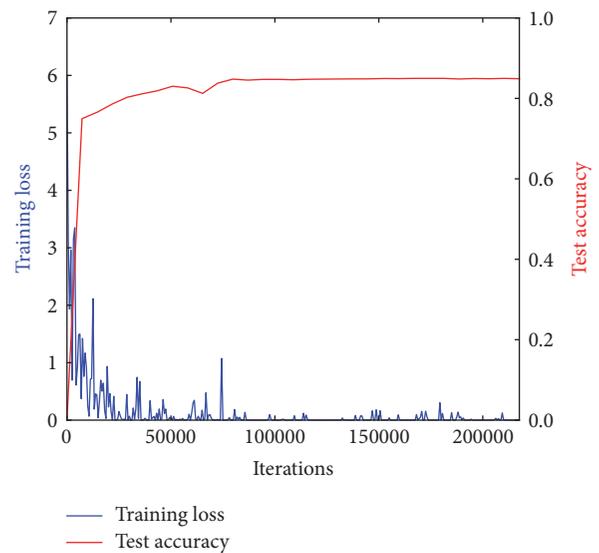


FIGURE 11: The training process of FTRN.

3.4. Experimental Result and Analysis on Logo-405. In this section, the results conducted on Logo-405 dataset by using three typical classifiers were reported, sequentially.

3.4.1. Results by Deep Architectures. We firstly listed the classification results by adopting fine-tuning deep architectures, as shown in Table 1.

The learning rate curves for the test accuracy and training loss of four fine-tuning CNNs were demonstrated in Figures 8–11, where the blue curve indicates the training loss rate and the red curve indicates the test accuracy.

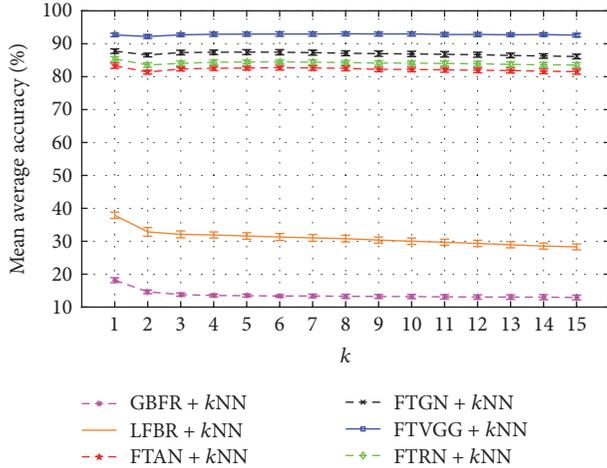
As can be seen, in general, FTVGG achieved convergence faster than three others. In terms of test accuracy, all of them

produced a dramatic increase at first, followed by a slight increase, and reached a steady state finally.

3.4.2. Classification Results by Combining Deep Representation and Traditional Classifiers. We conducted the classification tasks by combining deep representations and traditional classifiers. In this work, we adopted three typical classifiers, that is, *k*NN, *random forest*, and SVM. Since there are four deep representations obtained by fine-tuning deep CNN architectures, totally twelve different experimental combinations are produced.

TABLE 2: Classification comparison on SVM classifier with several strategies.

Approaches	MAA (%)	SD
GFBR	22.9685	0.8023
LFBR	65.8460	0.9976
FTAN + SVM	92.4142	0.3392
FTGN + SVM	91.0578	0.5200
FTVGG + SVM	95.9215	0.3597
FTRN + SVM	89.3848	0.5858

FIGURE 12: k NN classification performance comparison between different strategies.

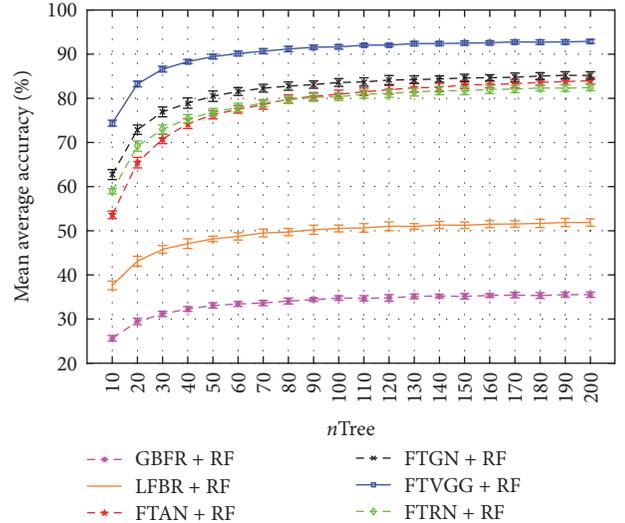
(1) *Results by Combining Deep Representation and k NN Classifier.* We conducted the k NN classification task with GFBR, LFBR, FTVGG + k NN, FTGN + k NN, FTAN + k NN, and FTRN + k NN in terms of 15 different values of k (the number of the nearest-neighbors), which differs from 1 to 15.

Figure 12 provides a graphical display of the experimental results with different representation strategies under different values of k . Both the MAA and SD of accuracy are illustrated in the results.

The results of Figure 12 demonstrate that (1) the approaches which combine both fine-tuning deep representation and k NN classifier, that is, FTVGG + k NN, FTGN + k NN, FTAN + k NN, and FTRN + k NN, consistently outperform the methods that adopt hand-crafted features, including GFBR and LFBR and (2) nearly all the strategies are not sensitive to the value of k , especially when k is greater than 4.

(2) *Results by Combining Deep Representation and Random Forest Classifier.* This section provides experimental results conducted on a *random forest* classifier with different strategies, that is, GFBR, LFBR, FTVGG + *random forest*, FTGN + *random forest*, FTAN + *random forest*, and FTRN + *random forest*. Experiments were carried out with 20 values of $nTree$ (the number of trees for random forest classifier), differing from 10 to 200.

Figure 13 provides a graphical display of the experimental results with different representation strategies under different

FIGURE 13: *Random forest* classification performance comparison between different strategies.

values of $nTree$, where RF indicates *random forest* classifier. Similarly, both the MAA and SD of accuracy are illustrated in the results.

We notice that (1) with respect to all the strategies, the performance apparently tends to be better when $nTree$ increases and (2) the performance of the approaches that combine fine-tuning deep representation and *random forest* classifier, that is, FTVGG + *random forest*, FTGN + *random forest*, FTAN + *random forest*, and FTRN + *random forest*, is significantly superior to LFBR and GBFR.

(3) *Results by Combining Deep Representation and SVM Classifier.* This section provides experimental results conducted on SVM classifier with different strategies, that is, GFBR, LFBR, FTAN + SVM, FTGN + SVM, FTVGG + SVM, and FTRN + SVM.

Table 2 lists the experimental results with different representation strategies. Both the MAA and SD of accuracy are also illustrated in the results.

Similar conclusion can be drawn from Table 3 where the performance of the approaches that combine fine-tuning deep representation and SVM classifier, that is, FTAN + SVM, FTGN + SVM, FTVGG + SVM, and FTRN + SVM, is significantly superior to LFBR and GBFR.

Lastly, we conclude this section by reporting the best performance of each strategy to compare three groups of

TABLE 3: The comparison between different strategies at their best performance.

FTAN 77.9084	FTAN + k NN 83.3292	FTAN + RF 83.9748	FTAN + SVM 92.4142
FTGN 82.3639	FTGN + k NN 87.7491	FTGN + RF 85.2070	FTGN + SVM 91.0578
FTVGG 84.9192	FTVGG + k NN 92.6811	FTVGG + RF 92.9046	FTVGG + SVM 95.9215
FTRN 85.3234	FTRN + k NN 85.3964	FTRN + RF 82.4291	FTRN + SVM 89.3848
	GFBR + k NN 18.1854	GFBR + RF 34.7383	GFBR + SVM 22.9685
	LFBR + k NN 37.8982	LFBR + RF 51.8752	LFBR + SVM 65.846

strategies, including the approaches that adopt fine-tuning deep CNNs (i.e., FTAN, FTGN, FTVGG, and FTRN), the methods which combine fine-tuning deep architectures and traditional classifiers (i.e., FTVGG + k NN, FTGN + k NN, FTAN + k NN, FTRN + k NN, FTVGG + *random forest*, FTGN + *random forest*, FTAN + *random forest*, FTRN + *random forest*, FTAN + SVM, FTGN + SVM, FTVGG + SVM, and FTRN + SVM), and those strategies employing hand-crafted features (i.e., GFBR, LFBR). The comparison results are shown in Table 3, where RF represents *random forest* classifier.

We have the observations from Table 3 that the proposed mechanisms that combine fine-tuning deep architectures and traditional classifiers demonstrate the superiority compared with other two groups of approaches, including the ones that adopt fine-tuning deep architectures and hand-crafted ones. The proposed classification mechanism specially obtains 5.4%, 6.1%, and 14.5% improvement on k NN, *random forest*, and SVM, respectively, towards FTAN strategy. For FTGN, it obtains 5.4%, 2.8%, and 8.7% when combining k NN, *random forest*, and SVM, respectively. With regard to FTVGG, it improves 7.8%, 8.0%, and 11% on k NN, *random forest*, and SVM, respectively. However, there is little improvement for FTRN when combining traditional classifiers. For example, FTRN + SVM improves 4.1% while FTRN + k NN obtains only 0.1% improvements.

With respect to the three classifiers used in the experiments, we observe that SVM outperforms k NN and *random forest* in nearly all tasks. Several factors may have contributed to this result. First, Logo-405 is of the high-dimensional representation, where the feature dimension of each logo is as high as 4096 in our deep representation strategies. Second, Logo-405 belongs to small sample size data compared with other large-scale datasets, for example, ImageNet [19]. Last, Logo-405 is of balanced data to some extent, in which each class consists of several tens to a hundred of logo images. We know that SVM works well for such kind of data, while k NN and *random forest* do not.

3.5. Experimental Result and Analysis on FlickrLogos-32. In this section, we evaluated the proposed mechanism on FlickrLogos-32 [20]. The experimental results conducted

TABLE 4: Classification results by adopting fine-tuning deep architectures.

(%)	FTAN	FTGN	FTVGG	FTRN
MAA	82.5893	90.1786	91.5179	92.8571

on Logo-405 dataset by using three typical classifiers are reported, sequentially.

3.5.1. Results by Fine-Tuning Deep Architectures. We also firstly listed the classification results by adopting fine-tuning deep architectures, as shown in Table 4.

The learning rate curves for the test accuracy and training loss of four fine-tuning CNNs were demonstrated in Figures 14–17, where the blue curve indicates the training loss rate and the red curve indicates the test accuracy.

As can be seen from the above results that the training process on FlickrLogos-32 obtains faster convergence compared with Logo-45 probably because of its smaller size. In general, FTRN achieved convergence a litter slower than three others. In terms of test accuracy, all of them produced a dramatic increase at first, followed by small fluctuation, and reached a steady state finally.

3.5.2. Classification Results by Combining Deep Representation and Traditional Classifiers. Similarly, we conducted the classification tasks by combining deep representations and traditional classifiers. In this work, we adopted three typical classifiers, that is, k NN, *random forest*, and SVM. Since there are four deep representations by fine-tuning deep CNN architectures, totally twelve different experimental combinations are produced.

(1) Results by Combining Deep Representation and k NN Classifier. We conducted the k NN classification task with GFBR, LFBR, FTVGG + k NN, FTGN + k NN, FTAN + k NN, and FTRN + k NN in terms of 15 different values of k (the number of the nearest-neighbors), which differs from 1 to 15.

Figure 18 provides a graphical display of the experimental results with different representation strategies under different values of k . Both the MAA and SD of accuracy are illustrated in the results.

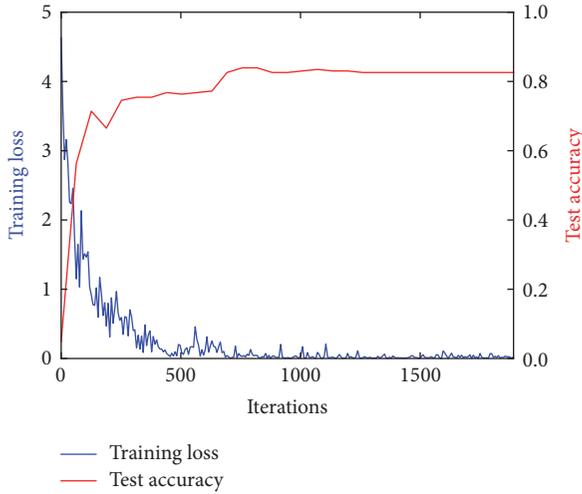


FIGURE 14: The training process of FTAN.

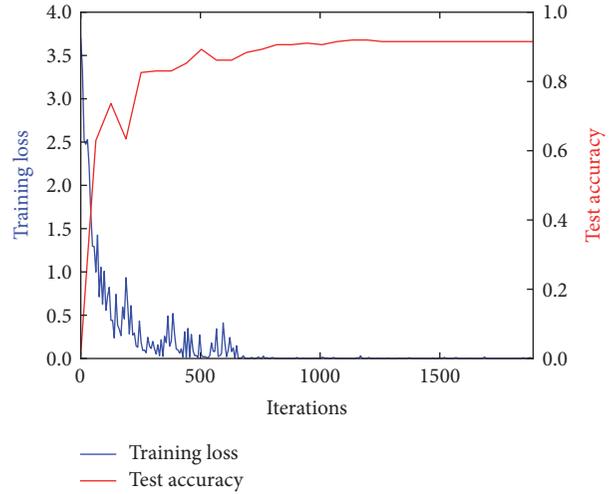


FIGURE 16: The training process of FTVGG.

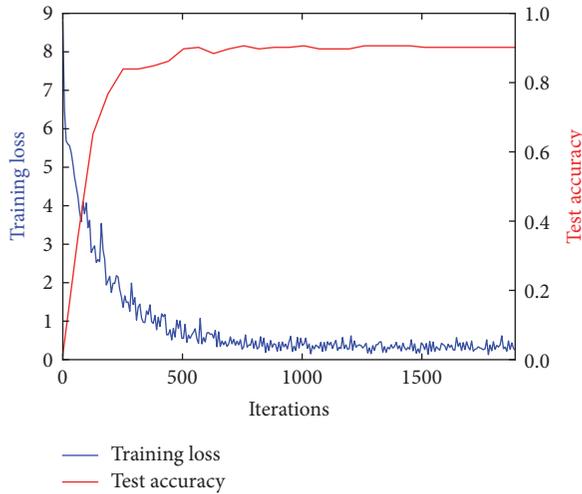


FIGURE 15: The training process of FTGN.

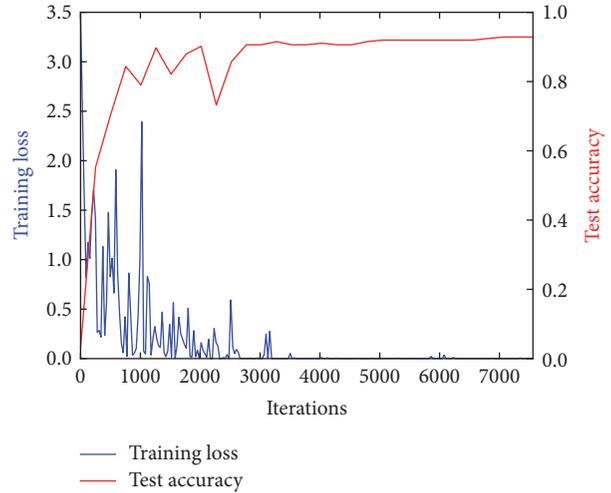


FIGURE 17: The training process of FTRN.

The results of Figure 18 demonstrate that (1) the approaches that combine both fine-tuning deep representation and k NN classifier, that is, FTVGG + k NN, FTGN + k NN, FTAN + k NN, and FTRN + k NN, consistently outperform the methods that adopt hand-crafted features, like GFBR and LFBR and (2) nearly all the strategies are not sensitive to the value of k , especially when k is greater than 3.

(2) Results by Combining Deep Representation and Random Forest Classifier. This section provides experimental results conducted on a *random forest* classifier with different strategies, that is, GFBR, LFBR, FTVGG + *random forest*, FTGN + *random forest*, FTAN + *random forest*, and FTRN + *random forest*. Experiments were carried out with 20 values of n_{Tree} (the number of trees for random forest classifier) differing from 10 to 200.

Figure 19 gives a graphical display of the experimental results with different representation strategies under different

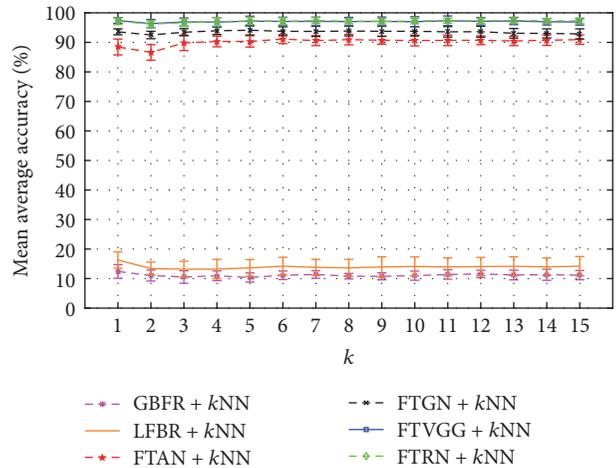


FIGURE 18: k NN classification performance comparison between different strategies.

TABLE 5: Classification comparison on SVM classifier with several strategies.

Approaches	MAA (%)	SD
GFBR	19.9107	2.6133
LFBR	72.1875	1.4888
FTAN + SVM	94.3304	1.3151
FTGN + SVM	94.7768	1.4125
FTVGG + SVM	98.1250	0.6243
FTRN + SVM	97.4554	1.0741

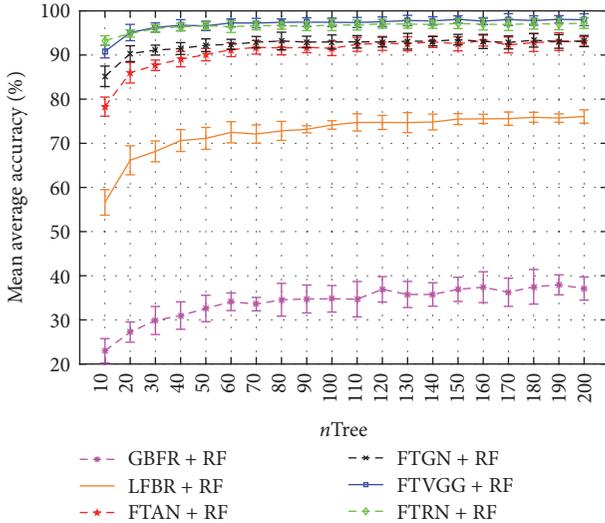


FIGURE 19: *Random forest* classification performance comparison between different strategies.

values of $nTree$. Similarly, both the MAA and SD of accuracy are illustrated in the results.

We find that, (1) with respect to all the strategies, the performance apparently tends to be better when $nTree$ increases and (2) the performance of the approaches that combine fine-tuning deep representation and random forest classifier, that is, FTVGG + *random forest*, FTGN + *random forest*, FTAN + *random forest*, and FTRN + *random forest*, is significantly superior to LFBR and GBFR.

(3) *Results by Combining Deep Representation and SVM Classifier*. This section provides experimental results conducted on SVM classifier with different strategies, that is, GFBR, LFBR, FTAN + SVM, FTGN + SVM, FTVGG + SVM, and FTRN + SVM.

Table 5 provides the experimental results with different representation strategies. Both the MAA and SD of accuracy are also illustrated in the results.

Similar conclusion can be draw from Table 5 where the performance of the approaches that combine fine-tuning deep representation and *random forest* classifier, that is, FTAN + SVM, FTGN + SVM, FTVGG + SVM, and FTRN + SVM, is significantly superior to LFBR and GBFR.

Lastly, we conclude this section by reporting the best performance of each strategy to compare three groups of strategies; they are (1) the approaches that adopt fine-tuning deep architectures (i.e., FTAN, FTGN, FTVGG, FTRN, and the method proposed by Bianco et al. in [13]), (2) the methods which combine fine-tuning deep architectures and traditional classifiers (i.e., FTVGG + *kNN*, FTGN + *kNN*, FTAN + *kNN*, FTRN + *kNN*, FTVGG + *random forest*, FTGN + *random forest*, FTAN + *random forest*, FTRN + *random forest*, FTAN + SVM, FTGN + SVM, FTVGG + SVM, and FTRN + SVM), and (3) those strategies employing hand-crafted features (i.e., GFBR, LFBR). The results are shown in Table 6, where RF represents *random forest* classifier.

We have the observations from Table 6 that the proposed classification mechanisms which combine fine-tuning deep architectures and traditional classifiers demonstrate the superiority compared with other two groups of approaches, including the ones that adopt fine-tuning deep architectures and hand-crafted ones. The proposed scheme specially obtains 8.5%, 10.7%, and 11.7% improvements on *kNN*, *random forest*, and SVM, respectively, towards FTAN strategy. With respect to FTGN, it obtains 3.9%, 3.3%, and 4.6% when combining *kNN*, *random forest*, and SVM, respectively. Regarding FTVGG, it improves 5.9%, 6.6%, and 6.6% on *kNN*, *random forest*, and SVM, respectively, while, with regard to FTRN, it can achieve 4.6%, 4.4%, and 4.6% improvements when combining *kNN*, *random forest*, and SVM, respectively. Compared to the method presented by Bianco et al. [13], the proposed mechanism obtains the improvement up to 7.125%.

4. Conclusion

With the amount of logo data on the Internet continuing to grow, designing effective management tools and systems is becoming imperative. This paper focuses on developing a fundamental tool for organizing logos by classifying them, which could make browsing and searching for logos more efficient. We design a combination mechanism that integrates both the advantages of deep learning models and traditional classification algorithms. Specifically, we firstly obtain the logo representations by fine-tuning several important deep architectures and then combine the learned logo representations with several traditional classifiers to carry out the logo classification task. While deep learning requires a large amount of data for training, we manage to achieve a high level of accuracy with a small-scale training set using transfer learning. Meanwhile, we build a Logo-405 dataset, which is larger than the existing logo datasets and can be publicly available. Experiments were conducted on both the Logo-405 dataset and FlickrLogos-32 dataset, and the results demonstrated that the proposed combination mechanism can effectively support logo classification and achieve better performance compared with other approaches, including the methods which integrate hand-crafted features and traditional pattern recognition algorithms and the models which employ deep CNNs.

TABLE 6: The comparison between different strategies at their best performance.

FTAN 82.5893	FTAN + k NN 91.1161	FTAN + RF 93.2589	FTAN + SVM 94.3304
FTGN 90.1786	FTGN + k NN 94.1071	FTGN + RF 93.4821	FTGN + SVM 94.7768
FTVGG 91.5179	FTVGG + k NN 97.4107	FTVGG + RF 98.0804	FTVGG + SVM 98.1250
FTRN 92.8571	FTRN + k NN 97.4107	FTRN + RF 97.2321	FTRN + SVM 97.4554
Bianco et al. [13] 91.0000	GFBR + k NN 12.4107	GFBR + RF 35.625	GFBR + SVM 19.9107
	LFBR + k NN 16.2946	LFBR + RF 74.1518	LFBR + SVM 72.1875

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was made possible through support from the major project of Natural Science Foundation of Shandong Province (ZR2016FQ20, ZR2014FM001), Postdoctoral Science Foundation of China (2017M612338), Natural Science Foundation of China (61702313, 61572300), Taishan Scholar Program of Shandong Province in China (TSHW201502038), and Fundamental Science and Frontier Technology Research of Chongqing CSTC (cstc2015jcyjBX0124).

References

- [1] A. P. Psyllos, C. N. Anagnostopoulos, and E. Kayafas, "Vehicle logo recognition using a sift-based enhanced matching scheme," *IEEE Transactions on Intelligent Transportation Systems*, vol. 11, no. 2, pp. 322–328, 2010.
- [2] Y. Gao, F. Wang, H. Luan, and T.-S. Chua, "Brand data gathering from live social media streams," in *Proceedings of the 2014 4th ACM International Conference on Multimedia Retrieval, ICMR 2014*, pp. 169–176, gbr, April 2014.
- [3] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: a review and new perspectives," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 8, pp. 1798–1828, 2013.
- [4] J. Wu, S. Pan, X. Zhu, C. Zhang, and X. Wu, "Positive and Unlabeled Multi-Graph Learning," *IEEE Transactions on Cybernetics*, vol. 47, no. 4, pp. 818–829, 2016.
- [5] J. Wu, S. Pan, X. Zhu, C. Zhang, and P. Yu, "Multiple structure-view learning for graph classification," *IEEE Transactions on Neural Networks and Learning Systems*, vol. PP, no. 99, pp. 1–16, 2017.
- [6] S. Hou, L. Chen, D. Tao, S. Zhou, W. Liu, and Y. Zheng, "Multi-layer multi-view topic model for classifying advertising video," *Pattern Recognition*, vol. 68, pp. 66–81, 2017.
- [7] J. Neumann, H. Samet, and A. Soffer, "Integration of local and global shape analysis for logo classification," *Pattern Recognition Letters*, vol. 23, no. 12, pp. 1449–1457, 2002.
- [8] S. Sun and Z. Chen, "Robust logo recognition for mobile phone applications," *Journal of Information Science and Engineering*, vol. 27, no. 2, pp. 545–559, 2011.
- [9] N. V. Kumar, V. V. Kantha, K. Govindaraju, and D. Guru, "Features fusion for classification of logos," *Procedia Computer Science*, vol. 85, pp. 370–379, 2016.
- [10] J. Wu, S. Pan, X. Zhu, C. Zhang, and X. Wu, "Multi-Instance Learning with Discriminative Bag Mapping," *IEEE Transactions on Knowledge and Data Engineering*, vol. 2017, pp. 1–16, 2017.
- [11] N. V. Kumar, V. V. Pratheek, K. N. Govindaraju, and D. S. Guru, "Features Fusion for Classification of Logos," *Procedia Computer Science*, vol. 85, pp. 370–379, 2016.
- [12] V. Nair and G. E. Hinton, "Rectified linear units improve Restricted Boltzmann machines," in *Proceedings of the 27th International Conference on Machine Learning (ICML '10)*, pp. 807–814, Haifa, Israel, June 2010.
- [13] S. Bianco, M. Buzzelli, D. Mazzini, and R. Schettini, "Deep learning for logo recognition," *Neurocomputing*, vol. 245, pp. 23–30, 2017.
- [14] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proceedings of the 26th Annual Conference on Neural Information Processing Systems (NIPS '12)*, pp. 1097–1105, Lake Tahoe, Nev, USA, December 2012.
- [15] C. Szegedy, W. Liu, Y. Jia et al., "Going deeper with convolutions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '15)*, pp. 1–9, Boston, Mass, USA, June 2015.
- [16] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proceedings of the 3th International Conference on Learning Representations (ICLR '15)*, pp. 1–14, May 2015.
- [17] S. Haykin and B. Kosko, *GradientBased Learning Applied to Document Recognition*, Wiley-IEEE Press, 2009.
- [18] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016*, pp. 770–778, usa, July 2016.
- [19] J. Deng, W. Dong, and R. Socher, "ImageNet: a large-scale hierarchical image database," in *Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 248–255, Miami, Fla, USA, June 2009.
- [20] S. Romberg, L. G. Pueyo, R. Lienhart, and R. Van Zwol, "Scalable logo recognition in real-world images," in *Proceedings of*

the 1st ACM International Conference on Multimedia Retrieval, ICMR'11, Trento, Italy, April 2011.

- [21] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.
- [22] H. Zhang, L. Cao, and S. Gao, "A locality correlation preserving support vector machine," *Pattern Recognition*, vol. 47, no. 9, pp. 3168–3178, 2014.
- [23] B. Gu, V. S. Sheng, and S. Li, "Bi-parameter space partition for cost-sensitive SVM," in *Proceedings of the 24th International Conference on Artificial Intelligence (ICAI '15)*, pp. 3532–3539, Las Vegas, Nev, USA, July 2015.

Research Article

A Distributed K -Means Segmentation Algorithm Applied to *Lobesia botrana* Recognition

José García,^{1,2} Christopher Pope,¹ and Francisco Altimiras^{1,3}

¹Telefonica Investigación y Desarrollo, Santiago, Chile

²Pontificia Universidad Católica de Valparaíso, 2362807 Valparaíso, Chile

³Faculty of Engineering and Sciences, Universidad Adolfo Ibáñez, Santiago, Chile

Correspondence should be addressed to José García; joseantonio.garcia@telefonica.com

Received 1 May 2017; Accepted 4 July 2017; Published 9 August 2017

Academic Editor: Jia Wu

Copyright © 2017 José García et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Early detection of *Lobesia botrana* is a primary issue for a proper control of this insect considered as the major pest in grapevine. In this article, we propose a novel method for *L. botrana* recognition using image data mining based on clustering segmentation with descriptors which consider gray scale values and gradient in each segment. This system allows a 95 percent of *L. botrana* recognition in non-fully controlled lighting, zoom, and orientation environments. Our image capture application is currently implemented in a mobile application and subsequent segmentation processing is done in the cloud.

1. Introduction

The grapevine moth or *Lobesia botrana* (Lepidoptera: Tortricidae) is an invasive insect species considered as one of the most damaging pests in vineyards (*Vitis vinifera* L.) [1]. Vineyards damage is caused by direct larval feeding inside grape berries, producing rot and dehydration, inducing lower productive yields and increasing the susceptible to other diseases such as gray and black mold (*Botrytis cinerea* and *Aspergillus niger*) [2, 3]. The *L. botrana* is widely distributed in Europe and is considered as a major grapevine pest since the twentieth century. In South America, *L. botrana* was first detected in Chile, in the area of Linderos, metropolitan region in April, 2008 [4]. Today, *L. botrana* is spread to all grape growing regions of Chile.

L. botrana management has become particularly relevant in Chile due to the potential economical impact as the main grape exporter worldwide. The moth control is performed using traps containing pheromones which attracts the moth males [5]. These traps are located on vineyards within an area formed of radius of 500 meters from a detected outbreak [6]. Currently, there are about 33.000 traps distributed throughout Chile for the specific control of *L. botrana*. To calibrate and verify moth growth models, in order to take measures

focused on the vineyards with outbreaks, it is needed to have information on the number of moths caught in each trap, in different time windows. However, the collection of this information is done manually, making the system costly in terms of time and not allowing to perform real-time actions on these vineyards. Therefore, it is needed to have a control system for automatically detection of *L. botrana*, to estimate the real number of the moths and to store the information associated with each trap for epidemiological studies. Given the number of traps and images captured, the system must have the ability to recognize specimens of *L. botrana* in large volumes of images. Additionally, because the jobs are executed in a specific period of the day, it is useful to manage the computing capacity according to demand.

One of the most satisfactory image recognition systems rely on binary classification using a sliding window approach. However, the sliding window approach strongly increases the computational cost, because the classifier function has to be evaluated over a large set of candidate subwindows [7]. Considering the need for massive image processing for the recognition of *L. botrana* specimens, this article proposes a novel method through a mobile application that captures images using a cell phone camera, together with an algorithm based on image segmentation, which performs

fast and automatic recognition of *L. botrana* specimens. The algorithm is based on using the k -means clustering technique together with gray scale and gradient descriptors to perform the classification [8]. Because the normal operation of the *L. botrana* recognition system may need to process over 30,000 images per day, a distributed version of the algorithm was implemented using Apache Spark as a distribution framework.

Experiments were performed where a version of the algorithm was evaluated using sliding window with the version that uses segmentation. The experiments evaluate the quality of the recognition and the execution times of both algorithms. For the algorithm that uses segmentation, we also evaluated the scalability of the distributed version. The results show that the segmentation algorithm has improvements to the execution time and the quality of the recognition with respect to the one that uses sliding window. The remainder of this paper is organized as follows. In Section 2 we present related work in agriculture and segmentation. The application that captures the images and stores them in the database is described in Section 3. A brief description of the distribution framework is done in Section 4. Section 5 describes the details of the segmentation algorithm together with its distributed version. Finally results and conclusions are described in Sections 6 and 7.

2. Related Work

2.1. Computer Vision in Agriculture. Computer vision has been used in agriculture and food processing over the last decades, with the aim to automate a variety of processes such as quality inspection, classification, and sorting of products [9]. The idea was to replace traditional manual operations that are labor intensive, slow, and prone to human error [10]. Despite the significant advances over the past decade, the creation of new algorithms, new paradigms, and new challenges to processing big complexity data allows applying computer vision research in real world problems, with a particular focus in agriculture [11–14].

One of these challenges is the application of image segmentation techniques, which separates the product region from background in image processing. It is one of the first steps in image analysis after the image capture to subdivide an image into meaningful regions. The segmentation result affects the subsequent image analysis. Image segmentation techniques in agriculture have been applied to the estimation of product size [15], shape [16], sorting [17], weed detection for site-specific treatment [18], and ground classification [19].

Another important area of research in the field of computer vision and agriculture has been the application of deep learning techniques [20–22]. This machine learning technique is based on the concept of neural networks and convolution to learn image features automatically by repeated training, error propagation, and incremental learning [20, 23–25]. Some deep learning architectures have been tested to identify moths and other insect pictures using images obtained mainly in laboratory conditions with great success [26–30].

2.2. Segmentation Techniques. In the next section we present the main image segmentation techniques currently being used.

(1) Edge Based Image Segmentation. Edge detection includes a variety of mathematical methods that is aimed at identifying points in a image at which the brightness changes sharply or has discontinuities. Using spectral methods and watershed morphological algorithms in [31], a segmentation framework based on edge detection was proposed. In [32] an edge detection based on illumination invariant feature detector phase congruency was proposed. The authors show that use of phase congruency for marking features has significant advantages over gradient-based methods. In [33] an edge based auto threshold select method to generate multiscale image segmentation was proposed. Band weight and Normalized Difference Vegetation Index (NDVI) are used to calculate edge weight.

(2) Threshold-Based Image Segmentation. In classical threshold image segmentation, an image is usually segmented and simply sorted to object and background by setting a threshold. But if there is complex information in the image, the threshold is not simple to obtain. An important line of research in threshold-based image segmentation has been to design algorithms that allow obtaining the optimum threshold. For example, in [34] a method to detect threshold based on entropy and applied to images in scale of grays was developed. Histograms were used in [35] to address the automatic detection of a threshold. In [36] a threshold-based level set approach including threshold-based segmentation and fast marching method was proposed. It was applied to medical image segmentation. In [37] they presented a new threshold segmentation method based on computational intelligence to improve the acquisition of images process in computer vision.

(3) Partial Differential Equation Based Image Segmentation. One way to perform segmentation is to detect the contours of the objects lying in the original image. The main idea of the PDE model is to translate the problem of segment objects into minimizing an energy function of a close curve. In literature there are numerous examples that apply PDE to segmentation. In [38] a variation model is presented using 4th-order PDE with 2nd-order PDE for the removal of finger vein image. In [39] a new segmentation model was used based on the geodesic contour model applied to color images. A new nonlinear partial differential equation (PDE) applied to gray images was proposed by [40].

3. Image Capture Application

A mobile and web application was developed in order to capture images and report relevant information to users. Figures 1 and 2 show screen shots for the mobile and web application. The mobile application is a native Android app. Images were taken on-field where traps hang, recording GPS position and date time of the capture. The image is then uploaded via cellular network (or Wi-Fi if available) to a

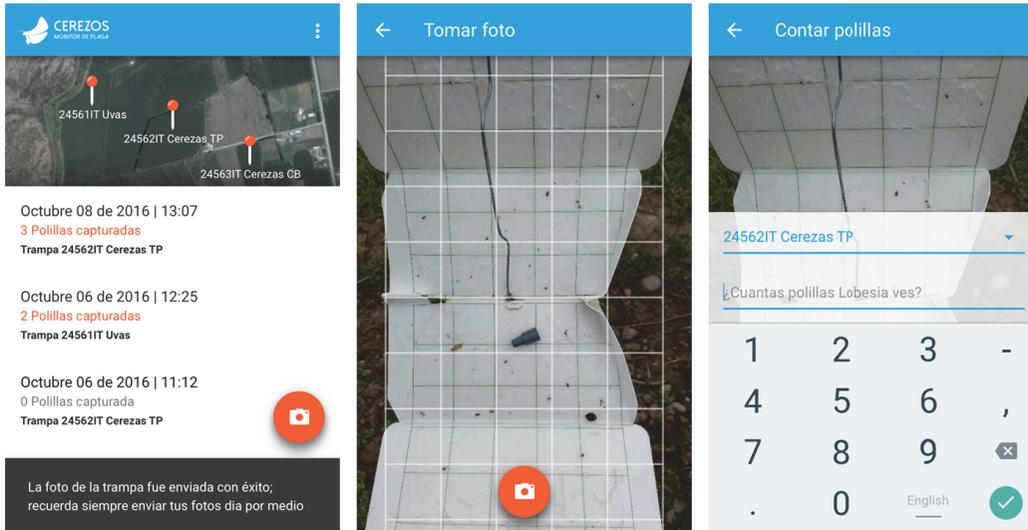


FIGURE 1: Mobile application.

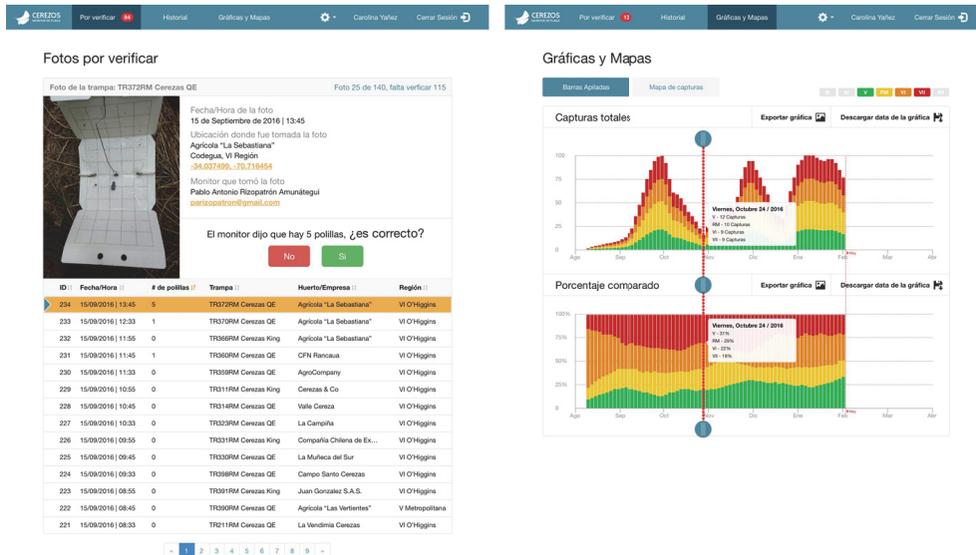


FIGURE 2: Web application.

cloud server to be processed. After processing, the image is displayed at the web application to an expert entomologist in order to assess the moth count and take action based on the pest situation for that specific place.

Figure 3 depicts the steps for capturing a trap image. First the trap must be opened and the sticky grid side must be placed facing forward to the capturing device. This device can be a cellular phone with a camera or a regular digital camera. Some traps have a triangular shape (delta traps) or a box shape. For some traps its sticky side has no grid, as shown in the example picture in Figure 3.

During September 2016 until February 2017, 50 traps were deployed between V, VI, VII, and metropolitan regions. A total of 26 users generated more than 400 on-field images. Figure 4 shows trap locations across V, VI, VII, and

metropolitan regions. These traps were visited regularly once a week and a picture was obtained each time. Figure 5 shows example of images obtained on-field using the developed mobile application.

4. Spark Distributed Framework

This section describes the main concepts of the distributed Spark framework. This framework will be used in Section 5.4 for the distributed version implementation of the segmentation algorithm. Distributed implementation aims to address the problem of large volumes of images.

In recent years the amount of available data has increased significantly [41]. This is mainly due to the simple and inexpensive storage process. However, this amount of data is

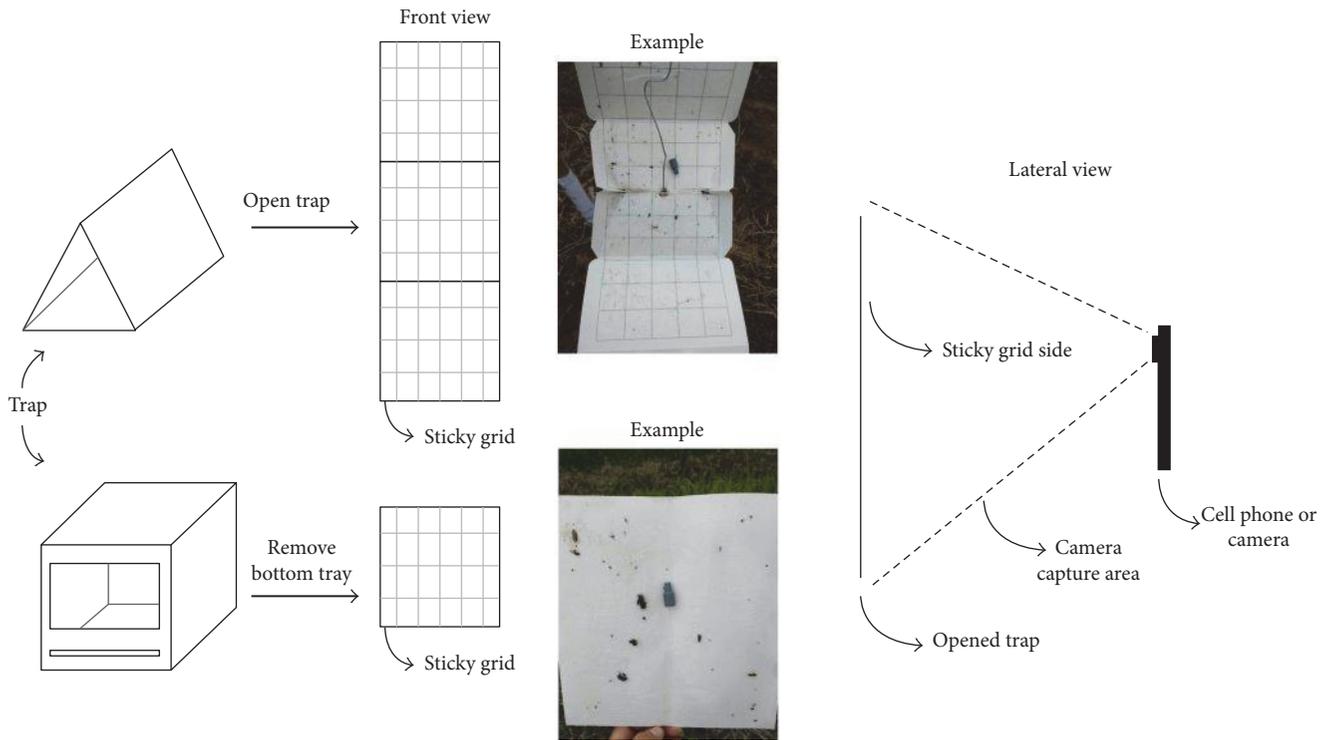


FIGURE 3: Image capture diagram.



FIGURE 4: Deployment map between V and VII regions for 50 traps, Chile. Source: Google Maps.

useless without an adequate process of extraction of knowledge where we use machine learning methods. This volume, diversity, and complexity [42] of the data brings challenges to researchers, because traditional machine learning methods can not deal with this situation. Cloud-based technologies provide an ideal environment to handle the data challenge. A pioneer in addressing this problematic of massive and complex data was the MapReduce Framework [43]. This

framework is based on the principle of locality of the data [44] which is implemented in a distributed filesystem. However, MapReduce is insufficient for applications that need to share data through multiple steps, or for iterative algorithms [45]. Many platforms for large-scale processing have recently emerged to overcome the issues presented by Hadoop MapReduce. Among them, Spark [46] appears as one of the most flexible and powerful engines to perform faster distributed computing in big data by using in-memory primitives.

Spark is a state of the art framework for high performance parallel computing designed to deal with iterative procedures that recursively perform operations over the same data. This has been used in machine learning algorithms [47], imaging processing [48], bioinformatics [49], computational intelligence [50], astronomy [51], medical information [52], and so on.

Spark was born as an in-memory cluster computing framework for processing and analyzing large amounts of data. It provides a simple programming interface, which enables an application developer to easily use the CPU, memory, and storage resources across a cluster of servers for processing large datasets in memory [46].

Spark has been positioned quickly as a general purpose platform. It provides a unified integrated platform for different types of data processing jobs. It can be used for batch processing, iterative process, interactive analysis, stream processing, machine learning, and graph computing.

Resilient Distributed Datasets (RDDs) are the core data units in Spark. These units are distributed and immutable;



FIGURE 5: Captures images examples.

that is, the transformation of RDDs is RDDs and fault-tolerant memory abstraction. There are two types of operations: transformations, which take RDDs and produce RDDs, and actions, which take RDDs and produce values. Various cluster management options can be used for run Spark, from simple Spark's standalone solutions, Apache Mesos, and Hadoop YARN [53].

Considering engineering applications we choose to use the Hadoop YARN management. Hadoop YARN, most recent implementation, uses cloud computing [54] which makes hundreds of machines provide services such as computing and storage on demand. Generally in-house implementations require large investments in hardware, software, and maintenance [54].

5. Algorithm

This section aims to describe the segmentation algorithm. First, a general description will be presented and then detailed by modules. The last section contains the distributed version of the algorithm.

Figure 6 shows the flowchart that allows constructing the classifier for *L. botrana* moths. As a dataset we consider 360 images obtained through the application described at Section 3. The 360 images correspond to three groups of 120 images each with resolutions 1280×720 , 1920×1080 , and 2048×1536 . 100 images are left to perform validation. For each one of the images a preprocessing, then a segmentation, and finally the generation of the descriptors that allow performing the training for the SVM classifier are applied. For the classifier training, a set of 5018 segments without moths and 2136 segments with moths were selected.

Additionally with the goal of evaluating the segmentation algorithm with respect to the recognition of *L. Botrana* we compared the performance to different elements usually glued on the sticky floor of the trap like other insects, leaf pieces, and equipment used in the trap. With this purpose, we analyzed 5018 segments without moths and a subset of 1325 segments containing elements other than the bottom of the trap were extracted.

5.1. Preprocessing Stage. As the first activity of the preprocessing stage, a Median Filter is used with the intention of removing particles of dust and brightness that appear at the photographs. An example is shown in Figure 7. The filter used was a 4×4 matrix where each pixel has the same weight. Subsequently an equalization is applied using the Contrast Limited Adaptive Histogram Equalization (CLAHE) algorithm. This algorithm uses histograms computed over different tile regions of the image. Local details can therefore be enhanced even in regions that are darker or lighter than most of the image. Then we apply 3 scaling ratios to the image: 0.75, 0.5, and 0.25. These scaling steps aim to consider that photographs can be taken at different distances from the cell phone. To perform the scaling, nearest-neighbor interpolation was used. Finally a gray scale conversion was executed. This conversion uses the ratio $0.3R + 0.6G + 0.1B$.

5.2. K-Means Segmentation Algorithm. The main objective of the project is to recognize *L. botrana* moths. A fundamental stage corresponds to the segmentation of moths. To perform the segmentation, each color image is scaled at 0.75, 0.50, and 0.25 and *K*-means clustering technique is applied to each image. In the definition of metric space to be able to apply *K*-means, the pixel distance shown in (1) and color distance shown in (2) are used. To achieve the proper segmentation a trade-off between pixel distance and color distance is realized. This trade-off is defined in (3). This type of segmentation has previously been used by [55] in benchmark datasets and in biomedical applications:

$$d_{xy} = \sqrt{(i_x - i_y)^2 + (j_x - j_y)^2}, \quad (1)$$

$$d_{\text{Lab}} = \sqrt{(L_x - L_y)^2 + (a_x - a_y)^2 + (b_x - b_y)^2}, \quad (2)$$

$$D_s = \sqrt{(d_{\text{Lab}})^2 + \left(\frac{m}{S}\right)^2 (d_{xy})^2}, \quad \text{where } S = \sqrt{\frac{N}{K}}. \quad (3)$$

D_s corresponds to the metric used to perform the segmentation and S is a normalizing factor, where N is the

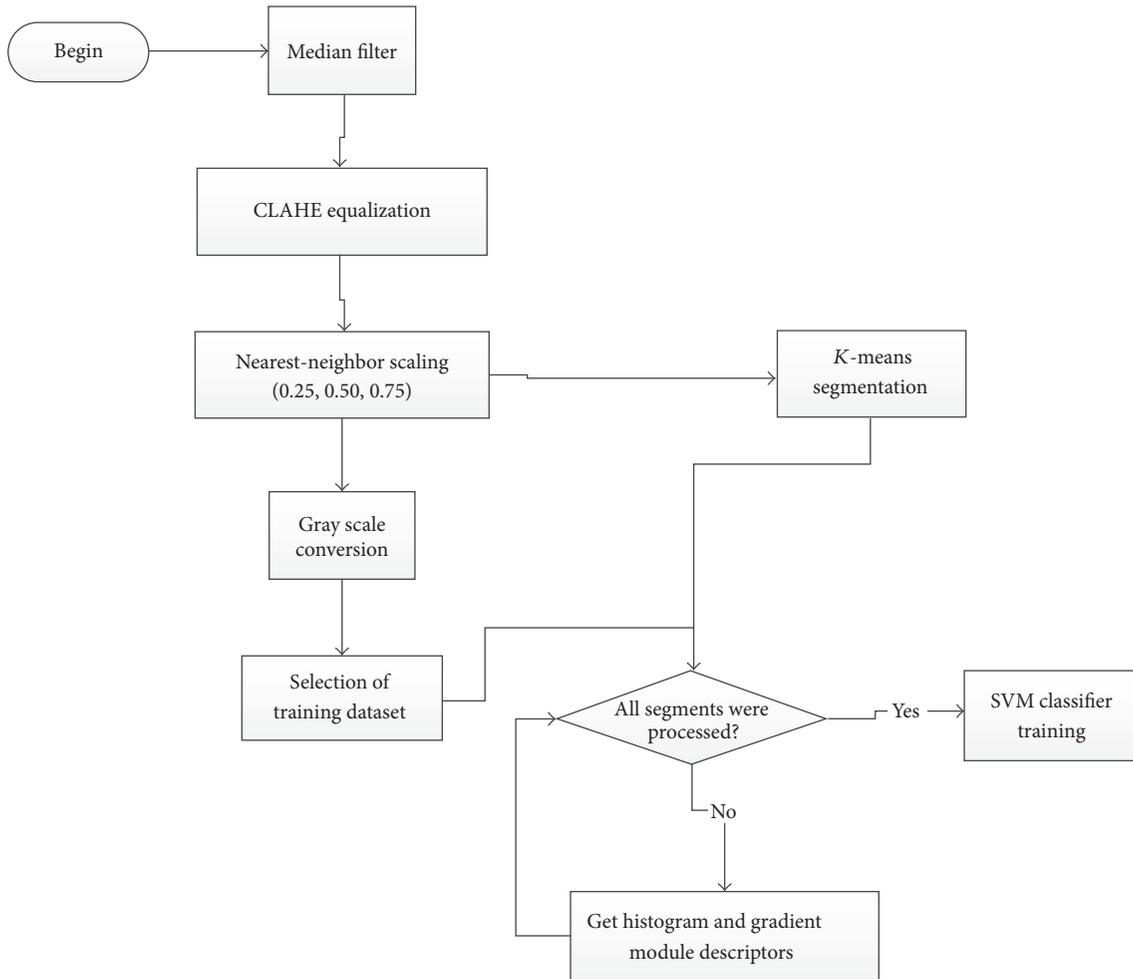


FIGURE 6: *Lobesia botrana* recognizer algorithm flow diagram.



FIGURE 7: (b) The original image, where powder can interfere with the histogram. (a) The processed image with equalization and median filter applied; the dust particles disappear.

number of pixels and K is the number of segments. The parameter m allows handling the weighting between the spatial distance and the colors distance. In Figure 8 it is possible to observe two segmentations. Figure 8(a) uses $K = 500$ and Figure 8(b) uses $K = 1000$.

5.3. Descriptors and SVM Classifier. After segmentation was performed, the next step is to differentiate the segments containing moths from those that do not include moths. To achieve this classification, gray scale segments are used. For each segment two descriptors are defined. The first

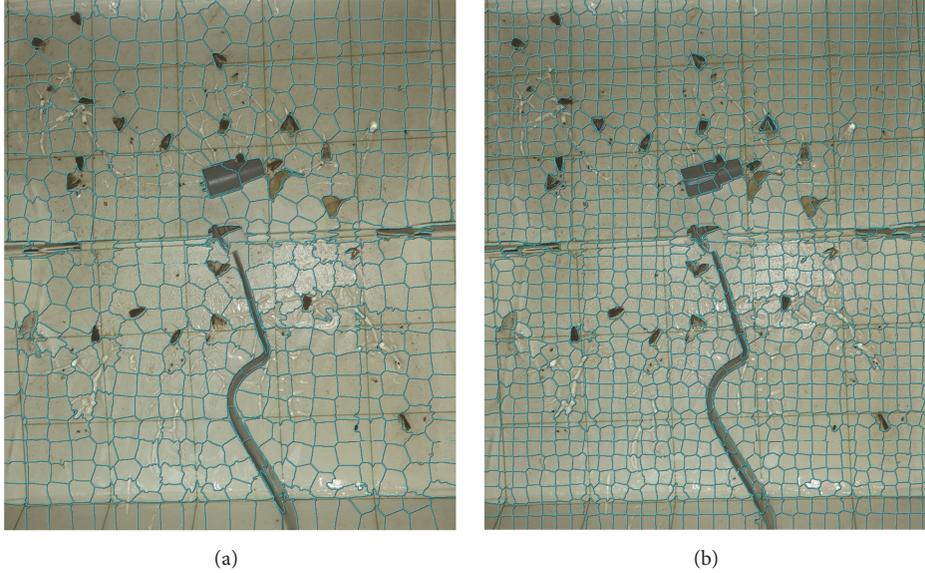


FIGURE 8: (a) corresponds to a segmentation considering 500 segments. (b) corresponds to segmentation considering 1000 segments.

descriptor corresponds to the histogram of the gray values of the segment. The histogram is configured with 10 bins and each bin is normalized with the amount of pixels that the segment contains. This generates a 10-dimensional vector where each dimension has values in the interval $[0, 1]$. A second descriptor is obtained as a result of calculating the histogram on the gradient modulus. This histogram is also normalized by the total number of pixels in the segment. In this case the number of bins used is 5. The following steps are used to calculate the gradient modulus.

(1) *Gradient Calculation.* A Sobel [56] operator is used. The complete image, I , is used to calculate each component of the gradient in all pixels of the image:

$$G_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 3 \end{bmatrix} * I, \quad (4)$$

$$G_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} * I.$$

(2) *Calculation of the Gradient Modulus.* For all pixels in image I the gradient modulus is calculated according to

$$G = \sqrt{G_x^2 + G_y^2}. \quad (5)$$

(3) *Histogram Calculation.* The histogram of G is calculated for each segment obtained from the K -means segmentation process.

The last step corresponds to the training the classifier. At training, 5018 segments are used without moths and 2136 segments with moths are considered. A K -fold cross validation was performed with $K = 5$ to perform the parameter adjustment.

5.4. *Distributed Algorithm.* Due to the estimated volume of photographs that need to be processed and considering that the process of segmentation along with the generation of descriptors corresponds to the one who consumes the most time, in this section we present a distributed version of the algorithm for segmentation and generation of descriptors. This version was designed using the Spark distributed computing framework.

The algorithm can be divided into two main stages: a segmentation stage and a descriptor generation stage. In the segmentation stage the algorithm operates on the complete image. The result of this stage corresponds to a list of segments. The descriptor generation stage has as input the list of segments and, for each one of these segments, descriptors are calculated. The distributed algorithm will address the distribution of calculations in the segmentation process as well as the calculation of descriptors in the segment list.

Segmentation Stage. Let us assume that the list of images has been read from Hadoop distributed file system (HDFS), used to store images. To perform the reading, the `binaryFiles` function is used, generating the corresponding RDD files. In addition, the list c of initial centroids has been obtained, considering an equidistant distribution for the different centroids. The distribution process in the segmentation stage is performed at the pixel and centroid level, where a pixel corresponds to the 5-dimensional distance-color vector defined in (3). The first stage of the segmentation phase is to perform a mapping shown in Algorithm 1. Each pixel i is mapped to the pair $(closestC, i)$ where $closestC$ corresponds to the centroid closest to pixel i . To perform this operation we use the Spark map transformation. The second stage is to perform a `reduceByKey`, where the key corresponds to the centroid $closestC$ (see Algorithm 2). The vector sum of the pixels is done in the 5-dimensional space, together with the count of the pixels associated with the centroid. To perform this

```

(1) Input Pixel  $i$ ,  $ListCentroids$ 
(2) Output Pixel  $i$ ,  $ClosestC$ 
(3) for each  $c$  in  $ListCentroids$  do
(4)   if  $D_s(p, c) < MinDist$  then
(5)      $MinDist \leftarrow D_s(p, c)$ 
(6)      $ClosestC \leftarrow c$ 
(7)   end if
(8) end for
(9) return ( $closestC, i$ )

```

ALGORITHM 1: Map segmentation function.

```

(1) Input keyValue  $closestC$ , Pixel  $i$ 
(2) Output  $total$ ,  $count$ 
(3) for each  $c$  in  $closestC$  do
(4)    $total \leftarrow total + i$ 
(5)    $count \leftarrow count + 1$ 
(6) end for
(7) return ( $total, count$ )

```

ALGORITHM 2: ReduceByKey segmentation function.

```

(1) Input segment  $s$ 
(2) Output segment  $s$ , descriptor  $d$ 
    $d \leftarrow getDescriptor(s)$ 
(3) return ( $s, d$ )

```

ALGORITHM 3: Map descriptor function.

operation, consider the reducebykey action. Finally the value of the new centroid is updated.

Descriptor Generation Stage. This stage aims to perform the calculation of the descriptors that allows classification. At this stage the image is distributed at the segment level. To perform the distribution a map function is used that maps the segment to the pair (segment, descriptor). Each map involves the calculation on a segment (see Algorithm 3).

6. Results

6.1. Parameters Setting. Considering that each algorithm involved requires parametrization, in order to find the best parameters, it was considered to perform a k -fold cross validation for each of the combinations shown at the Range column in Table 1. In the cross validation, $K = 5$ was considered. As a measure of the best configuration, F_1 -score = $2((precision * recall)/(precision + recall))$ was used. In Table 1, final column shows the selected value for the configuration that yields the best accuracy.

In order to have a better clarity of the effect of the segmentation on the classification, the parameter $NumSegment$ was studied. In order to perform this study, an experiment

TABLE 1: Segmentation algorithm setting parameters.

Parameters	Description	Range	Final
GridSize	CLAHE grid size	[8×8, 16×16, 32×32]	16 × 16
clipLimit	Clipping limit histogram	[2, 3, 4]	3
Median Filter Grid	Medium filter grid size	[4 × 4, 5 × 5, 6 × 6]	4 × 4
NumSegment	Number of image segments for segmentation algorithm	[500, 1000, 1500]	1000
m	Distance-color trade-off	[10, 15, 20]	15
C	Penalty parameter C SVM	[0.1, 1, 10]	1

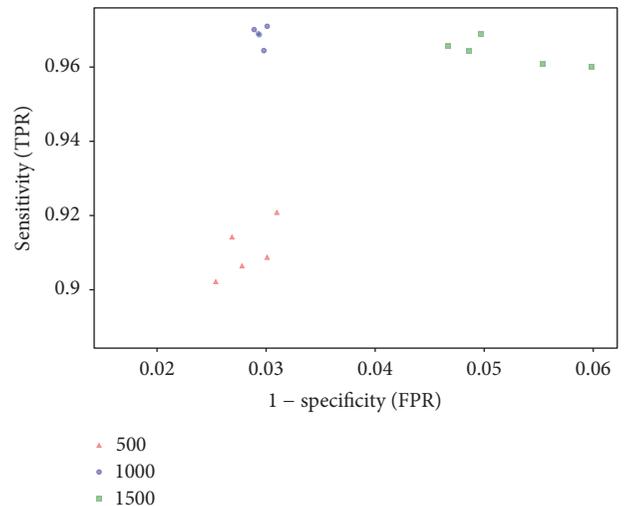
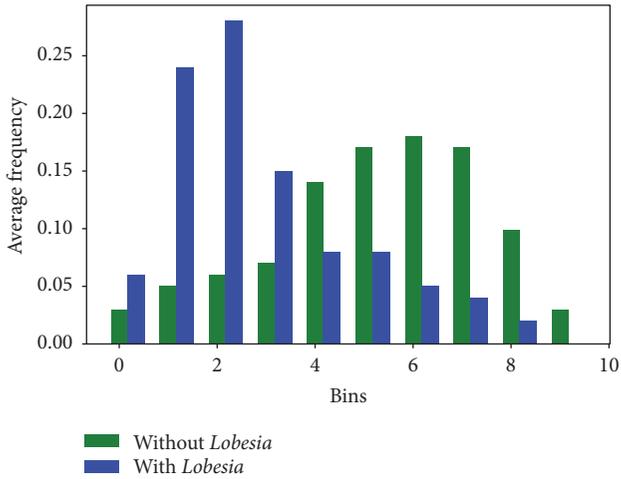
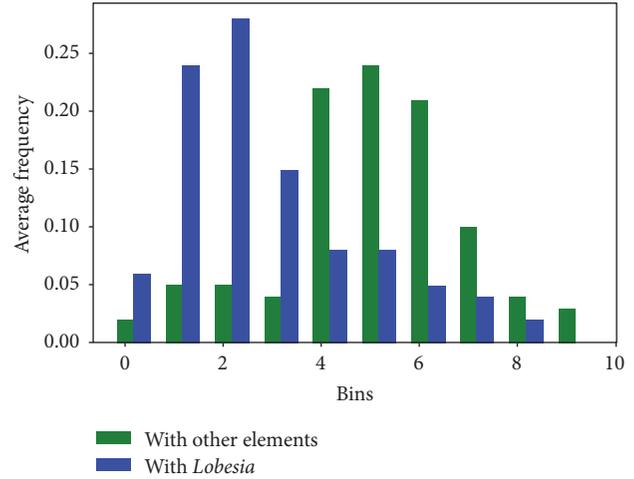
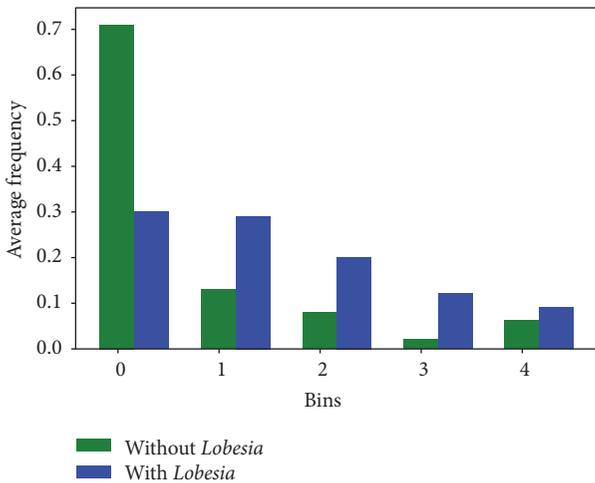
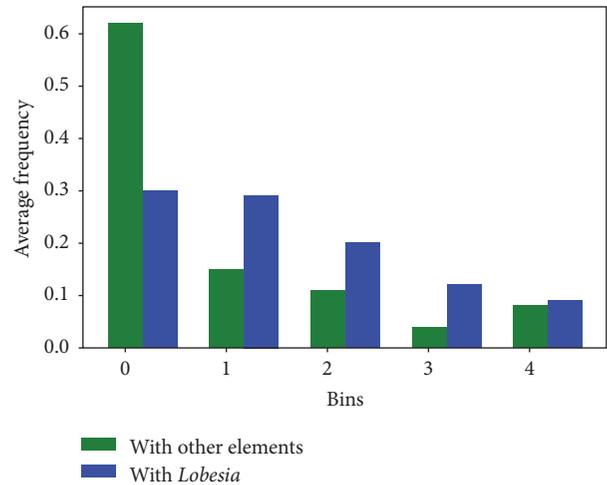


FIGURE 9: Sensitivity-specificity number segment evaluation.

was defined in which the remaining parameters are fixed considering the value of the final column and K -fold cross validation is performed with $K = 5$ considering the values 500, 1000, and 1500 in the parameter $NumSegment$. For each cross validation result, the True Positive Rate (TPR) and the False Positive Rate (FPR) were calculated. The result of the experiment is shown in Figure 9. It is observed that $NumSegment = 1000$, represented in blue color, corresponds to the best result, for the case $NumSegment = 500$ has a worse TPR performance and $NumSegment = 1500$ has a lower performance for FPR.

6.2. Descriptors Setting. As described at Section 5.3 a descriptor was used that combines a normalized histogram of gray scale with a normalized histogram using the gradient modulus. Equation (6) was used in order to determine the number of bins that best differentiates moth segments from those that do not contain moths

$$\text{Histogram Measure} = \sum_{i=1}^n \text{abs}(H_i(L) - H_i(WL)). \quad (6)$$

FIGURE 10: Gray scale descriptor *Lobesia* versus that without *Lobesia*.FIGURE 12: Gray scale descriptor *Lobesia* versus other elements.FIGURE 11: Gradient modulus descriptor *Lobesia* versus that without *Lobesia*.FIGURE 13: Gradient modulus descriptor *Lobesia* versus other elements.

L corresponds to the normalized average histogram of the segments with *L. botrana* and WL to the normalized average histogram of the segments without *L. botrana*. H_i corresponds to the function that returns the bin i of a histogram. For each bin, the value obtained from the *L. botrana* segments is subtracted to the value of segments without *L. botrana*. Then the modulus function and the sum on n bins are applied. The more alike the histograms are, the closer the *Histogram Measure* value is to 0; in our case we seek to maximize the *Histogram Measure* value. Ranges between 5 and 12 bins for the gray scale case and 4 to 10 bins for the gradient module were verified. The best results obtained after applying (6) are shown in Figures 10 and 11.

Finally, using the other elements dataset described in Section 5, we generate normalized average histograms for Gray and modulus of gradient. In Figures 12 and 13 the results are shown. Although the behavior of dataset histograms without *L. botrana* differs from histograms obtained by the

dataset with other elements, when comparing the latter with *L. botrana* histograms, an adequate separation is observed.

6.3. K-Means Segmentation Algorithm Evaluation. The evaluation of the segmentation algorithm was performed using the 100 images that were not included at the parameters configuration stage. To perform the comparisons an algorithm was designed using sliding window. When we apply sliding window, it is assumed that objects have regular appearance that do not deform much. A database is built with $n \times m$ fixed size windows containing the centered object and others that do not contain it. This is used to train the classifier. A $n \times m$ window is slid across the entire image and the classifier is queried if each of these windows contains or not the object. The sliding algorithm uses the same steps and configurations as the segmentation algorithm described at Section 5.2, except in the *K*-means segmentation stage where it uses sliding window. The window size was 64×64 and the step size was 8 pixels.

TABLE 2: Confusion matrix results.

Scale	Segmentation				Sliding window			
	TP	TN	FN	FP	TP	TN	FN	FP
1	4336	95174	191	299	4138	94805	389	668
0.75	4337	95120	190	353	4205	94996	322	477
0.50	4269	95062	257	411	4097	95187	430	286
0.25	4285	95082	242	391	4142	95091	385	382
Average	4307	92110	220	363	4145	95020	382	454

Comparisons were focused on the quality of the recognition as well as the execution speed of the different algorithms. In the case of the quality evaluation, the recognition for the rescaled images at 1, 0.75, 0.5, and 0.25 was analyzed. For the execution of the instances we use a PC running Windows 10, on an Intel® Core i7-4770 processor with 16 GB in RAM and programmed in Python 2.7.

Table 2 shows the resulting confusion matrix for both algorithms. It should be noted that the segmentation algorithm was superior to sliding window in all cases. Because our matrix has unbalanced class distribution, the following indicators were considered for comparison:

- (1) Sensitivity: $TP/(TP + FN)$
- (2) Precision: $TP/(TP + FP)$
- (3) F_1 Score: $2((precision * sencitivity)/(precision + sencitivity))$
- (4) Mathews correlation coefficient: $(TP * TN - FP * FN)/\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}$.

The segmentation algorithm yielded an average specificity of 95.1%; in the case of sliding window this was 91.4%. In the case of the precision indicator, the segmentation algorithm yielded an average value of 92.2%, and the sliding window was 90.01%. The F_1 score yielded by the segmentation algorithm was 93.65% and in the case of sliding window was 90.76%. Finally for the Matthews coefficient the value was 93.36% for segmentation and 90.41% for sliding windows.

Additionally, a confusion matrix was constructed to evaluate the performance of the algorithm with respect to the other elements data (Table 3). This dataset was generated from the 100 images with 1135 segments containing other elements. The segmentation algorithm for the case of the other elements data, obtained an average specificity of 95.15%, an accuracy of 99.5%, an F_1 coefficient of 97.2, and a Matthews coefficient of 88.1%.

To perform the comparison of execution times, we considered the three groups of images with resolutions of 1280×720 , 1920×1080 , and 2048×1536 . For each of the images the processing time is obtained and represented by a box plot shown in Figure 14. In this figure it is observed that the segmentation algorithm has a better execution time than sliding window. For the case of the segmentation algorithm the average time was 31.8 (s) for the group 1280×720 , 51.5 (s) for the result 1920×1080 , and 118 (s) for 2048×1536 . When using sliding window the time was 316.5, 649.2, and 1515.5 respectively. Finally Figure 15 shows the classification result for two test images.

TABLE 3: Confusion matrix results with other elements.

Scale	Segmentation			
	True	False		
Ground Truth	1	True	4336	191
		False	21	1114
Ground Truth	0.75	True	4337	190
		False	22	1113
Ground Truth	0.5	True	4269	257
		False	17	1118
Ground Truth	0.25	True	4285	242
		False	20	1115
Ground Truth	Average	True	4306.8	220
		False	20	1115

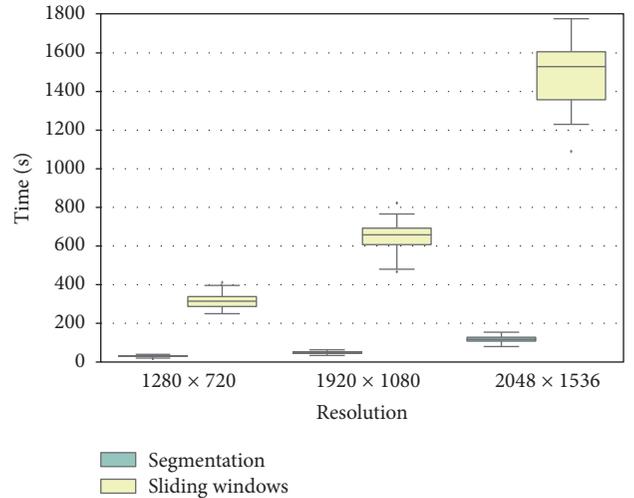


FIGURE 14: Time comparison using segmentation and sliding window techniques. Interquartile ranges: 25%, the median and 75% of the values.

6.4. K-Means Segmentation Algorithm Scalability. In the case of the distributed algorithm the experiments were focused on measuring its scalability. The algorithm was developed using Python 2.7 and Spark libraries. It was run on Azure platform using version 1.5.2 of Spark and Hadoop 2.4.1.

In order to evaluate the algorithm we considered the resolution, number of images, and different numbers of executors. As dataset the 360 images used to train and validate the algorithm were considered. These images were replicated



FIGURE 15: Example of processed images.

TABLE 4: Spark configuration.

num-executors	Executor-cores	Executor-memory (Gb)
1	3	4
2	3	4
4	3	4
8	3	4
16	3	4

1, 10, and 100 times in order to generate groups of 360, 3600, and 36000 images. Each group was divided according to their image resolution. For the scalability test 1, 2, 4, 8, and 16 executors were considered. The relative speed-up by group and resolution was measured. For the Spark configuration, three parameters were considered: num-executors which controls the number of executor requested, executor-cores property which controls the number of concurrent tasks an executor can run, and executor-memory which corresponds to the memory per executor. For the proper use of an executor it is recommended to use between 3 and 5 cores. The considered Spark settings are shown in Table 4.

Figure 16 shows the relative speed-up curves for Spark distributed implementation of segmentation algorithm. The relative speed-up is calculated with respect to the execution time obtained when performing the calculation using 1 executor. The blue line corresponds to the perfect result. Curves show at the three experiments that they continue delivering speed-up to 16 executors. When over 16 executors the performance is clearly sublinear. For the dataset with 360 images the deviation from the perfect result on average was 20.6%, for the case of 3600 images 25.6%, and for the case of 36000 images 32.1%. Regarding resolutions, the one that had the best performance in all cases was the one with the lowest resolution, but the effect of the resolution was small.

7. Conclusions

In this work we have presented a segmentation algorithm to perform the early recognition of *Lobesia botrana*. A mechanism for the adequate parameter setting selection and definition of the descriptors that allow classification

was developed. Image capture mobile and web application was created for *Lobesia botrana* recognition. The native Android mobile application takes the images and records the GPS position and date, making possible the real-time moth counting. The image recognition system was validated during September 2016 until 2017, in 50 traps deployed in vineyards between V, VI, VII, and metropolitan region in Chile.

The classifier algorithm developed for *L. botrana* recognition included a preprocessing step using a median filter and CLAHE equalization and then a segmentation step using *K*-means clustering and the generation of two descriptors; one corresponds to the gray scale segment histogram and the other obtained calculating the gradient modulus histogram. A dataset of 360 images was used for classifier construction.

The *K*-means segmentation algorithm was evaluated in comparison with the sliding window approach. The segmentation algorithm was superior to sliding window in quality of the recognition for the rescaled images at 1, 0.75, 0.5, and 0.25. The segmentation algorithm yielded an average specificity of 95.1.

A Spark distributed version of the system was developed. This system allows fast and scalable cloud-computing analysis of the images, providing an ideal environment for on-field applications. The scalability of the distributed *K*-means segmentation algorithm was evaluated. The curves show they continue delivering speed-up to 16 executors. When over 16 executors the performance is clearly sublinear. For the dataset with 360 images the deviation from the perfect result on average was 20.6%, for the case of 3600 images 25.6%, and for the case of 36000 images 32.1%. The one that had the best performance in all cases was the one with the lowest resolution, but the effect of the resolution was small.

Using the additional information of timestamps and GPS positioning captured by the implemented application, future work is intended to generate models that allow determining areas of risk of *Lobesia botrana* in addition to conditions that favor the development of the plague. From the point of view of computer science it is interesting to work in a computational intelligence algorithm that allows determining the best parameters setting. This automatic determination of parameters would allow the algorithm to be easily adaptable to similar recognition problems.

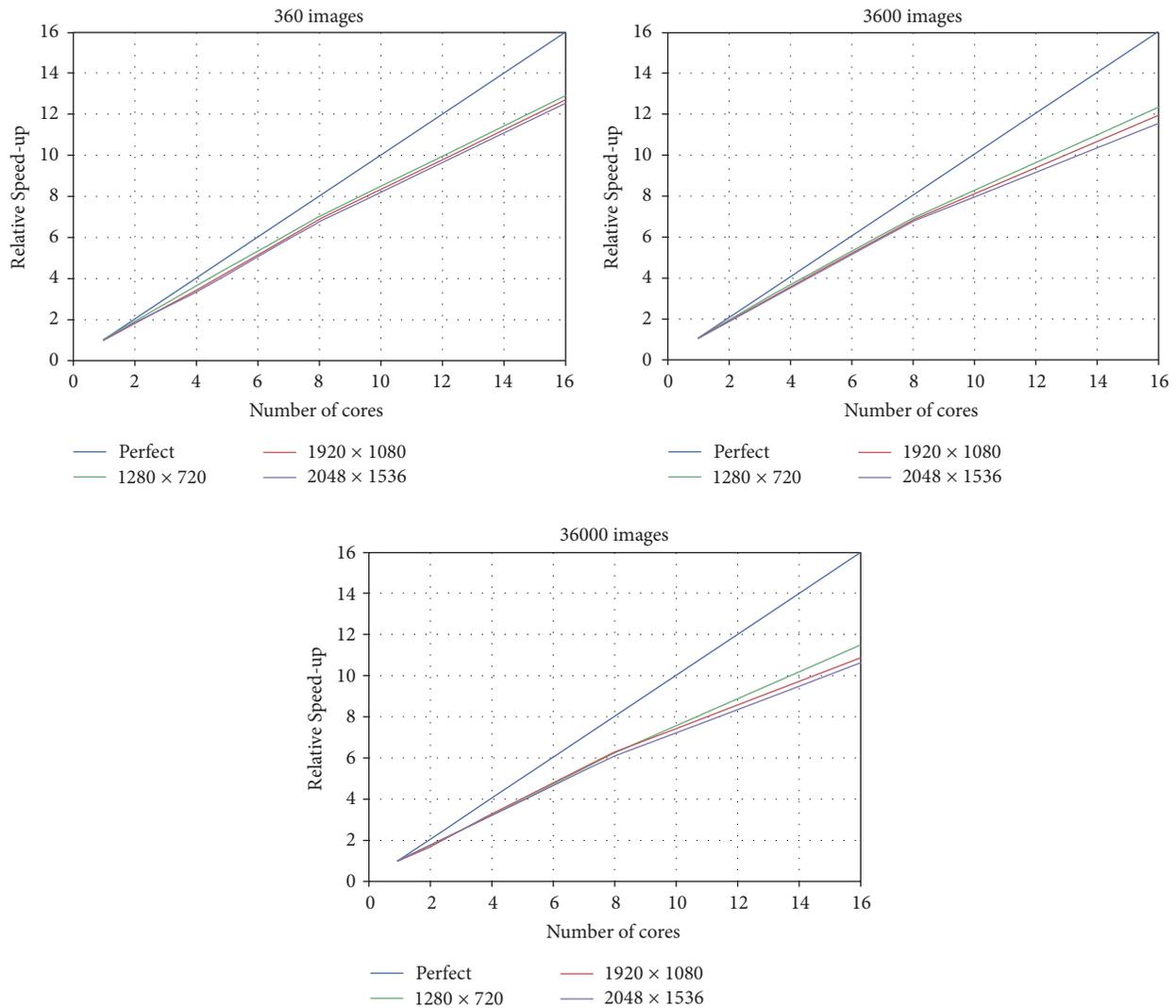


FIGURE 16: Relative speed-up experiments.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

- [1] J. Moreau, C. Villemant, B. Benrey, and D. Thiéry, "Species diversity of larval parasitoids of the European grapevine moth (*Lobesia botrana*, Lepidoptera: Tortricidae): The influence of region and cultivar," *Biological Control*, vol. 54, no. 3, pp. 300–306, 2010.
- [2] G. Cozzi, M. Pascale, G. Perrone, A. Visconti, and A. Logrieco, "Effect of *Lobesia botrana* damages on black aspergilli rot and ochratoxin A content in grapes," *International Journal of Food Microbiology*, vol. 111, pp. S88–S92, 2006.
- [3] M. Fermaud and Z. Giboulot, "Influence of *lobesia botrana* larvae on field severity of botrytis rot of grape berries," *Plant Disease*, vol. 76, no. 4, pp. 404–409, 1992.
- [4] C. Ioriatti, A. Lucchi, and L. G. Varela, "Grape berry moths in Western European Vineyards and their recent movement into the new world," in *Arthropod Management in Vineyards*, pp. 339–359, Springer, Dordrecht, The Netherlands, 2012.
- [5] C. Ioriatti, G. Anfora, M. Tasin, A. De Cristofaro, P. Witzgall, and A. Lucchi, "Chemical ecology and management of *lobesia botrana* (Lepidoptera: Tortricidae)," *Journal of Economic Entomology*, vol. 104, no. 4, pp. 1125–1137, 2011.
- [6] Servicio Agrícola y Ganadero (SAG), Programa nacional de *lobesia botrana*. Estrategia 2016-2017, 2016.
- [7] C. H. Lampert, M. B. Blaschko, and T. Hofmann, "Beyond sliding windows: object localization by efficient subwindow search," in *Proceedings of the 26th IEEE Conference on Computer Vision and Pattern Recognition (CVPR '08)*, pp. 1–8, IEEE, June 2008.
- [8] S. Ray and R. H. Turi, "Determination of number of clusters in *k*-means clustering and application in colour image segmentation," in *Proceedings of the 4th International Conference on Advances in Pattern Recognition and Digital Techniques*, pp. 137–143, 1999.
- [9] J. F. S. Gomes and F. R. Leta, "Applications of computer vision techniques in the agriculture and food industry: A review,"

- European Food Research and Technology*, vol. 235, no. 6, pp. 989–1000, 2012.
- [10] D.-W. Sun, *Computer Vision Technology for Food Quality Evaluation*, Academic Press, 2016.
 - [11] S. Benalia, S. Cubero, J. M. Prats-Montalbán, B. Bernardi, G. Zimbalatti, and J. Blasco, “Computer vision for automatic quality inspection of dried figs (*Ficus carica* L.) in real-time,” *Computers and Electronics in Agriculture*, vol. 120, pp. 17–25, 2016.
 - [12] S. R. Debats, D. Luo, L. D. Estes, T. J. Fuchs, and K. K. Caylor, “A generalized computer vision approach to mapping crop fields in heterogeneous agricultural landscapes,” *Remote Sensing of Environment*, vol. 179, pp. 210–221, 2016.
 - [13] J. Ma, D.-W. Sun, J.-H. Qu et al., “Applications of computer vision for assessing quality of agri-food products: a review of recent research advances,” *Critical Reviews in Food Science and Nutrition*, vol. 56, no. 1, pp. 113–127, 2016.
 - [14] M. Nagle, I. Kiatkamjorn, G. Romano, B. Mahayothee, V. Sardasud, and J. Müller, “Determination of surface color of “all yellow” mango cultivars using computer vision,” *International Journal of Agricultural and Biological Engineering*, vol. 9, no. 1, article 42, 2016.
 - [15] G. P. Moreda, J. Ortiz-Cañavate, F. J. García-Ramos, and M. Ruiz-Altisent, “Non-destructive technologies for fruit and vegetable size determination—a review,” *Journal of Food Engineering*, vol. 92, no. 2, pp. 119–136, 2009.
 - [16] G. P. Moreda, M. A. Muñoz, M. Ruiz-Altisent, and A. Perdigones, “Shape determination of horticultural produce using two-dimensional computer vision - A review,” *Journal of Food Engineering*, vol. 108, no. 2, pp. 245–261, 2012.
 - [17] A. Mizushima and R. Lu, “An image segmentation method for apple sorting and grading using support vector machine and Otsu’s method,” *Computers and Electronics in Agriculture*, vol. 94, pp. 29–37, 2013.
 - [18] X. P. Burgos-Artizzu, A. Ribeiro, A. Tellaeché, G. Pajares, and C. Fernández-Quintanilla, “Improving weed pressure assessment using digital images from an experience-based reasoning approach,” *Computers and Electronics in Agriculture*, vol. 65, no. 2, pp. 176–185, 2009.
 - [19] J. D. Luscier, W. L. Thompson, J. M. Wilson, B. E. Gorham, and L. D. Dragut, “Using digital photographs and object-based image analysis to estimate percent ground cover in vegetation plots,” *Frontiers in Ecology and the Environment*, vol. 4, no. 8, pp. 408–413, 2006.
 - [20] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.
 - [21] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” in *Proceedings of the 26th Annual Conference on Neural Information Processing Systems (NIPS ’12)*, pp. 1097–1105, Lake Tahoe, Nev, USA, December 2012.
 - [22] LISA Lab University of Montreal, *Deep Learning Tutorial*, Release 0.1, 2013.
 - [23] D. C. Cireşan, U. Meier, L. M. Gambardella, and J. Schmidhuber, “Convolutional neural network committees for handwritten character classification,” in *Proceedings of the 11th International Conference on Document Analysis and Recognition*, pp. 1135–1139, September 2011.
 - [24] M. D. Zeiler and R. Fergus, “Visualizing and understanding convolutional networks,” in *Computer Vision—ECCV 2014*, vol. 8689 of *Lecture Notes in Computer Science*, pp. 818–833, Springer, 2014.
 - [25] J. Mairal, P. Koniusz, Z. Harchaoui, and C. Schmid, “Convolutional kernel networks,” in *Proceedings of the 28th Annual Conference on Neural Information Processing Systems (NIPS ’14)*, pp. 2627–2635, December 2014.
 - [26] C. Wen, D. Wu, H. Hu, and W. Pan, “Pose estimation-dependent identification method for field moth images using deep learning architecture,” *Biosystems Engineering*, vol. 136, pp. 117–128, 2015.
 - [27] J. L. Miranda, B. D. Gerardo, and B. T. Tanguilig III, “Pest detection and extraction using image processing techniques,” *International Journal of Computer and Communication Engineering*, vol. 3, no. 3, pp. 189–192, 2014.
 - [28] W. Ding and G. Taylor, “Automatic moth detection from trap images for pest management,” *Computers and Electronics in Agriculture*, vol. 123, pp. 17–28, 2016.
 - [29] F. Zhang, G. Wang, and J. Ye, “An automatic identification system of agricultural pests based on machine vision,” *Journal of Computational and Theoretical Nanoscience*, vol. 12, no. 12, pp. 6231–6236, 2015.
 - [30] Y. Kaya and L. Kayci, “Application of artificial neural network for automatic detection of butterfly species using color and texture features,” *Visual Computer*, vol. 30, no. 1, pp. 71–79, 2014.
 - [31] F. C. Monteiro and A. Campilho, “Watershed framework to region-based image segmentation,” in *Proceedings of the 19th International Conference on Pattern Recognition (ICPR ’08)*, pp. 1–4, IEEE, December 2008.
 - [32] R. V. Patil and K. C. Jondhale, “Edge based technique to estimate number of clusters in k-means color image segmentation,” in *Proceedings of the 3rd IEEE International Conference on Computer Science and Information Technology (ICCSIT ’10)*, vol. 2, pp. 117–121, IEEE, July 2010.
 - [33] W. Cui and Y. Zhang, “Graph based multispectral high resolution image segmentation,” in *Proceedings of the International Conference on Multimedia Technology (ICMT ’10)*, pp. 1–5, IEEE, October 2010.
 - [34] A. S. Abutaleb, “Automatic thresholding of gray-level pictures using two-dimensional entropy,” *Computer Vision, Graphics and Image Processing*, vol. 47, no. 1, pp. 22–32, 1989.
 - [35] P.-Y. Yin and L.-H. Chen, “A new method for multilevel thresholding using symmetry and duality of the histogram,” in *Proceedings of the International Symposium on Speech, Image Processing and Neural Networks (ISSIPNN ’94)*, pp. 45–48, IEEE, Hong Kong, 1994.
 - [36] A. Xu, L. Wang, S. Feng, and Y. Qu, “Threshold-based level set method of image segmentation,” in *Proceedings of the 3rd International Conference on Intelligent Networks and Intelligent Systems (ICINIS ’10)*, pp. 703–706, IEEE, November 2010.
 - [37] W. Kaihua and B. Tao, “Optimal threshold image segmentation method based on genetic algorithm in wheel set online measurement,” in *Proceedings of the 3rd International Conference on Measuring Technology and Mechatronics Automation (ICMTMA 20)*, vol. 2, pp. 799–802, IEEE, January 2011.
 - [38] F. Zhang, S. Guo, and X. Qian, “Segmentation for finger vein image based on PDEs denoising,” in *Proceedings of the 3rd International Conference on BioMedical Engineering and Informatics (BMEI ’10)*, vol. 2, pp. 531–535, IEEE, October 2010.
 - [39] C. Yuan and S. Liang, “Segmentation of color image based on partial differential equations,” in *Proceedings of the 4th International Symposium on Computational Intelligence and Design (ISCID ’11)*, vol. 2, pp. 238–240, IEEE, October 2011.
 - [40] J. Xiao, B. Yi, L. Xu, and H. Xie, “An image segmentation algorithm based on level set using discontinuous pde,” in *Proceedings*

of the 1st International Conference on Intelligent Networks and Intelligent Systems (ICINIS '08), pp. 503–506, IEEE, November 2008.

- [41] C. Lynch, “Big data: how do your data grow?” *Nature*, vol. 455, no. 7209, pp. 28–29, 2008.
- [42] M. Minelli, M. Chambers, and A. Dhiraj, *Big Data, Big Analytics: Emerging Business Intelligence and Analytic Trends for Today's Businesses*, John Wiley & Sons, 2012.
- [43] J. Dean and S. Ghemawat, “MapReduce: simplified data processing on large clusters,” *Communications of the ACM*, vol. 51, no. 1, pp. 107–113, 2008.
- [44] Z. Guo, G. Fox, and M. Zhou, “Investigation of data locality in mapreduce,” in *Proceedings of the 12th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGrid '12)*, pp. 419–426, IEEE, May 2012.
- [45] K. Grolinger, M. Hayes, W. A. Higashino, A. L'Heureux, D. S. Allison, and M. A. Capretz, “Challenges for mapreduce in big data,” in *Proceedings of the IEEE World Congress on Services (SERVICES '14)*, pp. 182–189, IEEE, June 2014.
- [46] M. Zaharia, M. Chowdhury, M. J. Franklin, S. Shenker, and I. Stoica, “Spark: cluster computing with working sets,” in *Proceedings of the 2nd USENIX Conference on Hot Topics in Cloud Computing (HotCloud '10)*, p. 10, June 2010.
- [47] R. Shyam, S. Kumar, P. Poornachandran, and K. P. Soman, “Apache spark a big data analytics platform for smart grid,” *Procedia Technology*, vol. 21, pp. 171–178, 2015.
- [48] S. Sarraf and M. Ostadhashem, “Big data application in functional magnetic resonance imaging using apache spark,” in *Proceedings of the Future Technologies Conference (FTC '16)*, pp. 281–284, IEEE, December 2016.
- [49] W. Zhou, R. Li, S. Yuan et al., “MetaSpark: a spark-based distributed processing tool to recruit metagenomic reads to reference genomes,” *Bioinformatics*, vol. 33, no. 7, pp. 1090–1092, 2017.
- [50] J. García, B. Crawford, R. Soto, and P. García, “A multi dynamic binary black hole algorithm applied to set covering problem,” in *Proceedings of the International Conference on Harmony Search Algorithm*, vol. 514, pp. 42–51, Springer, 2017.
- [51] Z. Zhang, K. Barbary, F. A. Nothaft et al., “Kira: processing astronomy imagery using big data technology,” *IEEE Transactions on Big Data*, 2016.
- [52] L. Hao, S. Jiang, B. Si, and B. Bai, “Design of the research platform for medical information analysis and data mining,” in *Proceedings of the 9th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI '16)*, pp. 1985–1989, IEEE, Datong, China, October 2016.
- [53] H. Karau, A. Konwinski, P. Wendell, and M. Zaharia, *Learning Spark: Lightning-Fast Big Data Analysis*, O'Reilly Media, Inc., 2015.
- [54] D. Agrawal, S. Das, and A. El Abbadi, “Big data and cloud computing: current state and future opportunities,” in *Proceedings of the 14th International Conference on Extending Database Technology*, pp. 530–533, ACM, March 2011.
- [55] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk, “SLIC superpixels compared to state-of-the-art superpixel methods,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 11, pp. 2274–2281, 2012.
- [56] R. Boyle and R. Thomas, *Computer Vision: A First Course*, Blackwell Scientific Publications, 1988.

Research Article

Determining the Coupling Source on a Set of Oscillators from Experimental Data

Juan Carlos Jauregui-Correa,¹ Carlos S. López-Cajun,² and Mihir Sen³

¹*División de Investigación y Posgrado, Facultad de Ingeniería, Universidad Autónoma de Querétaro, 76010 Santiago de Querétaro, QRO, Mexico*

²*Departamento de Ingeniería Mecánica, Facultad de Ingeniería, Universidad Autónoma de Querétaro, 76010 Santiago de Querétaro, QRO, Mexico*

³*Department of Aerospace and Mechanical Engineering, University of Notre Dame, Notre Dame, IN 46556, USA*

Correspondence should be addressed to Juan Carlos Jauregui-Correa; jc.jauregui@uaq.mx

Received 9 January 2017; Accepted 5 April 2017; Published 4 May 2017

Academic Editor: Jia Wu

Copyright © 2017 Juan Carlos Jauregui-Correa et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Complex systems are a broad concept that comprises many disciplines, including engineering systems. Regardless of their particular behavior, complex systems share similar behaviors, such as synchronization. This paper presents different techniques for determining the source of coupling when a set of oscillators synchronize. It is possible to identify the location and time variations of the coupling by applying a combination of analytical techniques, namely, the source of synchronization. For this purpose, the analysis of experimental data from a complex mechanical system is presented. The experiment consisted in placing a 24-bladed rotor under an airflow. The vibratory motion of the blades was recorded with accelerometers, and the resulting information was analyzed with four techniques: correlation coefficients, Kuramoto parameter, cross-correlation functions, and the recurrence plot. The measurements clearly show the existence of frequencies due to the foreground components and the internal interaction between them due to the background components (coupling).

1. Introduction

Complex systems are common in engineering, examples being communication networks, computer networks, and electronic devices, as well as mechanical systems such as manufacturing plants, aircrafts, and automobiles. Determining the dynamic behavior of these mechanical systems requires the use of advanced analytical techniques. As the number of elements increases, simple systems evolve into ones that are more complex. Concepts such as correlation dimension, Lyapunov exponents, fractal properties, and complex network analysis are some of ideas that are used to determine dynamic properties. The problem is to identify the source of synchronization from field data.

In addition to other behavioral characteristics, complex systems can present synchronization. In most cases, the system displays two behaviors: one that is easily identified at a foreground level and another that remains at background

level [1]. The foreground behavior is seen at frequency responses with high amplitudes and narrow bands, whereas the background behavior presents a wide band response and low amplitudes. The study of synchronization has been an exciting and interesting problem since Huygens first wrote about his experiments in 1665 [2]. Peña Ramirez et al. [3] analyzed Huygens' experiment, modeling the system as a lumped mass model, and found that synchronization depends on the relationship between the masses, the stiffness of the connecting beam, and the amount of damping in the system.

Keeffe et al. [4] presented a model for the analysis of pulse coupled oscillators. As they described it: in some systems synchronization starts at a certain location and expands forming clusters. They suggested that clusters formed, evolved, and collapsed. For this purpose, they developed a model that provides an idea of the physical phenomena and how clusters evolved into synchronization. They described

different systems that behave as pulse coupled oscillators. They presented two types of oscillators, systems with local coupling and systems with global coupling. There is still a question how transient dynamics leads to synchronization [4]. Ulrichs et al. [5] presented the analysis of metronomes as nonlinear periodic oscillators. The metronomes behave asymptotically and they show higher dimensional attractors. Metronomes have the same behavior as Huygens clocks, and their synchronization can be modeled using Kuramoto's parameter. They found small traveling waves through the system support that lock the phase difference among all metronomes and they model the connections in terms of a Van der Pol oscillator. Aragonès and Guasch [6] presented a theoretical analysis of path propagation in a set of systems with similar structures. Woodhouse [7] and Langley et al. [8, 9] introduced statistical energy analysis. They modeled the system as a set of springs and represented it as a block diagram. The diagram represented subsystems as blocks and power flows as connections among subsystems. For the net analysis, they assigned weight factors relating a coupling loss factor to a total loss factor. Moreover, they dealt with the net analysis including a probabilistic density function and computed the weight factors as a statistical ratio. They defined the length of a particular link as a random variable, and they defined this function as a mean function plus a function of the standard deviation. A similar approach was presented by Llerena-Aguilar et al. [10]. They reported a hybrid method for the analysis of wave propagation and synchronization of a Wireless Acoustic Sensor (microphones) Networks. The problem they solved is the inverse of what we are trying to understand. In fact, they studied the desynchronization. The phenomenon is caused by two factors, namely, the clock phase offset of each sensor and the propagation delay due to the distance between them. These factors are critical in the selection of the processing algorithm. The classical solution was the implementation of clock synchronization protocols. The phenomenon is simulated assuming that the source signals are received by an array of sensors. Then, the signals are combined adding a Gaussian noise and convoluting the signal with an impulse function for each sensor. They discretized the signal using the Short Time Fourier Transform and they aligned the mixture of signals using the cross-correlation and the Log Energy Cross-Correlation functions. Ray et al. [11] investigated the Noise Induced Synchronization in two identical time delay systems. They compared numerical results with experimental data and they analyzed the effect of white, green, and red noises. Messina and Vittal [12] reported a method for analyzing dynamic patterns from measurements. They applied the Hilbert transform and proper orthogonal analysis technique. This technique is based on the relation of the energy and phase embedded into the mode function. Rosenblum and Pikovsky [13] presented a method for detecting weak couplings between two oscillators. If two oscillators are weakly coupled, they assume that the two signals contain enough data from the weak interaction. The relationship in a driven-response from noisy data could be unreliable and the interpretation is difficult. They discussed the effect of noise as follows: "in case of perfect synchrony, we are not able to separate the effect of interaction from

the internal dynamics of autonomous systems. In order to obtain the information on the direction of coupling we need to observe deviations from the synchrony, either due to noise or due to onset of quasiperiodic dynamics outside the synchronization region." They propose a method for determining the direction of the coupling and they quantified the cross-dependencies of the phase dynamics. They illustrate their method with two coupled Van der Pol oscillators.

Friedrich et al. [14] presented an approach for the study of complex systems using stochastic methods. Complex systems can be traced back to rather simple laws because of the existence of self-organized processes. While the analysis of nonlinear dynamics is traditionally conducted using time series, complex systems show stochastic data in time and length scales. It is clear that the synchronization of complex systems is a function of time and space. In Friedrich's work, the dynamic model is represented as a Langevin equation with white noise. Since the dynamic model can be represented as a deterministic set of nonlinear differential equations, they added a stochastic term with white noise. They considered a Gaussian distribution for time propagation as a Markov process. A model for self-similarity was included, meaning that at a certain time two elements should have the same statistics. In a previous work, Ghasemi et al. [15] presented the steps below for developing the stochastic equation. First, review if the data follow a Markov chain. Then, estimate the Markov time scale, which is the minimum time interval for a Markov process. For this analysis, it is necessary to use a joint probability density function. This can be simplified with the Chapman-Kolmogorov equation, which is the probability of having two similar distributions in a time interval. Then, they derived the stochastic equation and a regeneration of the stochastic process. They applied the method to heart beat signals. In complementary analyses, Siefert et al. [16] and Shirazi et al. [17] classified complex systems as networks with fluctuating stochastic processes. For this problem, they represented spatial networks of complex dynamics in phase space. They proposed a method for mapping a stochastic process onto a complex network. As for recurrence plots, they determined the Markov time scale, and they based their analysis on the Markov process.

Marwan et al. [18] presented a review of recurrence plots applied to dynamic systems. Recurrence is one of the basic properties of dynamic systems and the recurrence plots can help visualize dynamic system behavior. They described the use of recurrence plots for identifying synchronization in the sense that two systems synchronize if their recurrence in phase space is linearly dependent. In another work, Donner et al. [19, 20] studied the dynamics of complex systems using the recurrence concept and network theory. They demonstrated that exploiting the duality of the recurrence matrix and the adjacent matrix of a complex network improved the determination of quantitative characteristics of recurrence networks. They divided them into three types of recurrence networks, that is, local, intermediate, and global. All their results were obtained from simulations.

Other researchers have applied different techniques to identify synchronization. Huo et al. [21] analyzed the synchronization of a couple of calcium oscillators. They proposed

a mathematical model based on a set of nonlinear first-order differential equations. They found synchronization through the analysis of phase diagrams. Kim and Lim [22] worked on burst and spike synchronization of neurons. Their interest was aroused by the analysis of electroencephalogram (EEG) data. Researchers believed that synchronization of neuron activities could help in understanding sensory and cognitive processes. From an engineering point of view, it can be understood as periods of high frequency impulse generation and periods of no impulse generation. Xu et al. [23] and Rosário et al. [24] described motif synchronization. They studied temporal and spatial synchronization from EEGs and applied time-varying graphs, static network, and motif synchronization. Motif synchronization is a sequence of small patterns in a particular order of occurrence. With three first-order differential equations, they simulated the propagation of an external stimulus in time and space.

Childs and Strogatz [25] analyzed a periodically forced Kuramoto model. They defined a rich dynamic system when it has a strong interconnection among randomness, force, and coupling. Randomness is associated with variations in the natural frequencies; this concept is important because it determines the desynchronization of the oscillators. The coupling is related to the oscillator's phase and the force to the external excitation. Thus, the system synchronizes and desynchronizes depending on the relative magnitude of these three effects. Another important concept described by Childs and Strogatz is that a subset of the oscillators is “phase-locked” to the force drive, whereas the rest of the oscillators are on the tails of the frequency distribution outside of synchronization. Their method is applied to a set of Van der Pol oscillators and they proposed that the oscillators are sinusoidal coupling; in our case it is assumed that the coupling has a very low frequency compared to the natural frequency and the amplitude is low.

The objective of the present paper is to determine, using experimental data, if there is a way to identify the background effects that enable synchronization in complex systems and coupling. It is assumed that a coupling source exists and is the cause that synchronizes all the elements. The difficulty of identifying this coupling source rests on the fact that its dynamic response is of low frequency and low amplitude. First, different techniques that describe synchronization are applied; then, data evolution is analyzed in order to determine if there is a very slow dynamic behavior. The results show that synchronization is easily identified, but the source of synchronization is not; the reason is its low energy content and its larger fundamental frequency. Direct FFT, or other traditional dynamic analysis techniques, is unable to process this type of dynamics. The background modes are, however, responsible for internal interaction, communication, and energy transfer between the natural frequencies of the foreground components. The background vibrations are transmitted through somewhat undefined neighboring linkages at very low frequencies, which may be orders of magnitude lower than the natural frequencies of the foreground element.



FIGURE 1: Test rotor with 24 blades.



FIGURE 2: Experimental setup and laser vibrometer used for calibration.

2. Experimental Setup

The experimental setup consisted of a stationary rotor with 24 blades. In order to reduce the number of variables, the blades were built with a rectangular cross section, as shown in Figure 1, and they were mounted with the thin side of the blade facing the front. At the tip of each blade, a MEMS accelerometer was mounted; data from 22 accelerometers were recorded. These accelerometers were double-axis Analog Devices ADXL321 type with a sensitivity of 10 g. Their masses were small enough so they had a negligible contribution to the natural frequency of the blades. A laser Doppler vibrometer, similar to that used by Oberholster and Heyns [26], was used to calibrate the individual accelerometers that were mounted on each blade. Figure 2 shows the experimental setup and the laser vibrometer used for calibration. The accelerometers were connected to a data acquisition system, and the data were recorded simultaneously.

The rotor was tested under two types of loads, namely, an impact and an airflow test [27]. For the impact test, a calibrated hammer was used to hit the center of the rotor and all the blades vibrated simultaneously. Data were collected for 1.6 s at a sampling rate of 5000 Hz. The airflow test consisted of blowing air through a duct producing an airflow perpendicular to the blades. The flow velocity was kept constant and accelerometer data were recorded continuously over a time interval of 2 s at a sampling rate of 1 kHz. There

TABLE 1: Starting times for ten data recording events with airflow.

Event	Starting time
A	10:31 AM
B	10:43 AM
C	10:54 AM
D	11:06 AM
E	11:18 AM
F	11:29 AM
G	11:41 AM
H	11:52 AM
I	12:04 PM
J	12:16 PM

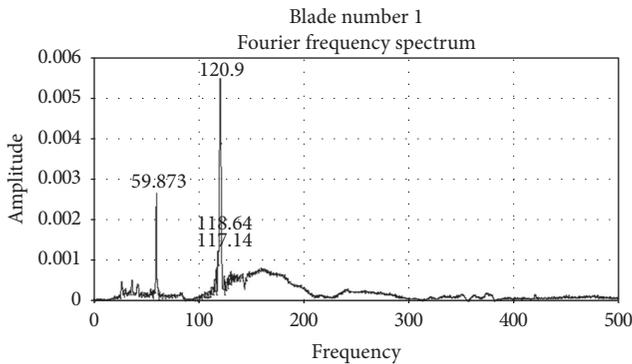


FIGURE 3: Fourier spectrum, impact test.

were ten recordings intervals, called events (Table 1). Event A started when the airflow was switched on.

3. Data Analysis

The natural frequencies of each blade were determined with an impact test. Figure 3 shows the Fourier spectrum of blade 1. It is clear that there are two dominant peaks related to the foreground components, 120.9 Hz corresponding to the natural frequency of the blade and 59.9 Hz to the rotor structure.

The blades were found to have almost the same natural frequency between 121 and 122 Hz. Since the present analysis is focused on determining the state of synchronization, for the airflow experiments each blade response was filtered around 120 Hz. Figure 4 shows the frequency spectrum of the airflow test corresponding to event A. In this way, only the natural dynamic response of the blades was included in the analysis. Filtering is an important step since other vibration sources hide synchronization, and the lock to the driven force is minimized [25]. It was found that the analysis techniques are very sensitive to external sources of vibration. The analysis presented in this paper is centered on the difference between each foreground element; therefore, if there is a strong source of vibration, every accelerometer will measure the same value and the contribution of the interaction between the background components will disappear.

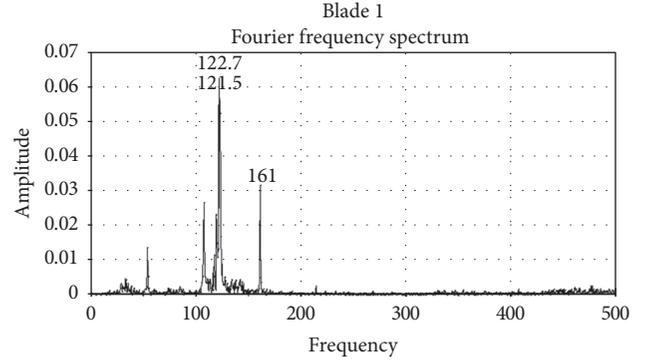


FIGURE 4: Fourier spectrum with air flow: blade 1, event A.

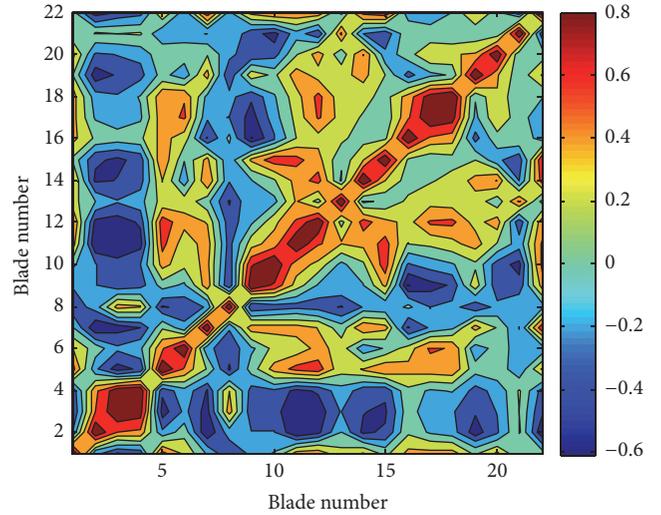


FIGURE 5: Correlation map for the impact test.

4. Correlation Coefficient

The correlation coefficient is a method that calculates the angle between two vectors. This coefficient calculates the inner product of the data vectors and measures the linear dependency of two signals. Although it differs from the phase angle, it gives a good estimation of the synchronization of two signals. Applying the Pearson correlation coefficient to the 22 data vectors a symmetric 22 by 22 matrix is obtained. Unfortunately, since this parameter measures the overall phase angle, it cannot identify the internal interaction that brings the elements into synchronization.

The correlation coefficient for the impact test is displayed in Figure 5. This figure presents a pattern that is intrinsic to the actual structure. Ideally, this figure must have the same values except at the diagonal, but the differences correspond to deviations in the construction of each blade. This figure is the characteristic signature of the structure.

Every map showed a similar pattern to the impact test. This pattern is associated with the rotor structure; it is a signature of the system and appears regardless of the analysis. As a result, the correlations between blade 5 and blades 13 and 16 are greater than 0.5 in every map. Similarly, blade

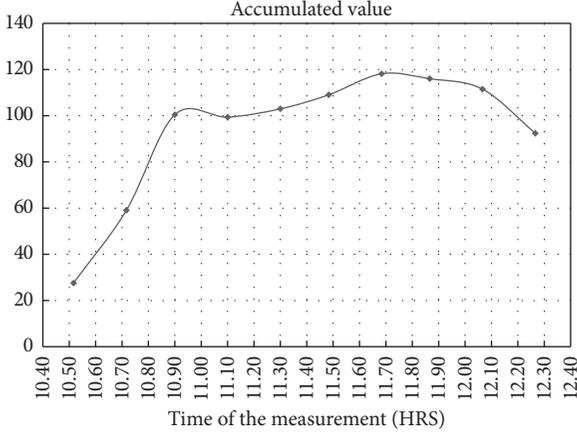


FIGURE 6: Magnitude of the correlation matrix as a function of real time.

10 has a strong correlation with blade 20. The rest of the blades have oscillatory values for the entire test. From these maps, it is difficult to define a function that describes these oscillations, since they vary in time and position within the matrix. Therefore, the magnitude of each matrix was evaluated. Figure 6 shows the variations of the accumulated value as a function of time:

$$\text{sum} = \sum_j \sum_i \rho_{ij}. \quad (1)$$

It can be seen that after event C there is little variation.

The correlation factor is unable to identify synchronization, and to overcome this the data were analyzed as indicated next.

5. Kuramoto Parameter

The Kuramoto model describes analytically the phase transition of a set of oscillators into a mutual synchronization and determines a threshold at which the oscillators are coupled [25]. The Kuramoto parameter measures the instantaneous phase angle of several signals. It is based on the analysis of the synchronization of a set of oscillators moving at a similar frequency but with a variable phase. This parameter was used to determine the amount of synchronization among the blades. The instantaneous phase angle was determined using the Hilbert transform. This is given by

$$\tilde{x}(t) = H[x(t)] = \int_{-\infty}^{\infty} \frac{x(u)}{\pi(t-u)} du, \quad (2)$$

where $x(t)$ and $\tilde{x}(t)$, both, are real-valued time domain signals.

From the Hilbert transform, a complex vector is defined as

$$z(t) = x(t) + i\tilde{x}(t). \quad (3)$$

Thus, the instantaneous phase angle is calculated as

$$\tan(\phi(t)) = \frac{\tilde{x}(t)}{x(t)}. \quad (4)$$

After calculating the instantaneous phase among the blades, the Kuramoto parameter was determined to be

$$r = \frac{1}{n} \left| \sum_{j=1}^n e^{i\phi_j(t)} \right|, \quad (5)$$

where $\phi_i(t)$, $i = 1, \dots, n$, represent the instantaneous phase angle.

When the Kuramoto parameter is equal or close to 1, all the oscillators move together. If it is equal or close to zero, then the oscillators move independently and no synchronization occurs. Since the actual data contain several frequencies, the Kuramoto parameter will evaluate the contribution of all frequencies, including the excitation frequency. In general, this frequency will dominate over the rest; thus, it was necessary to filter the response data around the natural frequency. With this procedure, the Kuramoto parameter was calculated at each event. To have an estimation of the degree of synchronization, the histogram of the Kuramoto parameter was calculated, and it was assumed that the blades are synchronized if the majority of the computed values are above 0.7. The Kuramoto parameter is a good indication of synchronization. Nevertheless, it is unable to identify the source of synchronization.

The result of analyzing the impact test is shown in Figure 7.

In every event, most of the time the Kuramoto parameter has a value around 0.7, indicating that the blades vibrate synchronously. Nevertheless, it is difficult to identify the source of synchronization from these plots.

6. Cross-Correlation Function

This technique of analysis consists of determining the cross-correlation function between the blades. This function is calculated between the acceleration data of one blade with respect to the other 21 blades. It estimates the statistical correlation between two signals at two different points in space or time. It is a useful indicator of dependencies or similarities in time of two signals. In addition, it can be used to interpolate for values at instants for which there are no observations. With this basis, the cross-correlation function determines similarities among the data and disregards those instances when the blades vibrate out of synchronization.

Mathematically, the cross-correlation function is defined as

$$C(\tau) = f * g = \int_{-\infty}^{\infty} f^*(t) g(t + \tau) dt, \quad (6)$$

where τ is defined as the lag.

Based on the elements of a rich dynamics, the internal interaction between the background components occurs in time and space; the cross-correlation function can be used to produce maps in time and space; time is represented by the lag order, being a normalized parameter that is proportional to time, and the blade number represents space. At each event interval, there are 22 maps. The maps are constructed by

$$C(n, \tau)_j = f_n * g_j = \int_{-\infty}^{\infty} f^*(t)_n g(t + \tau)_j dt, \quad (7)$$

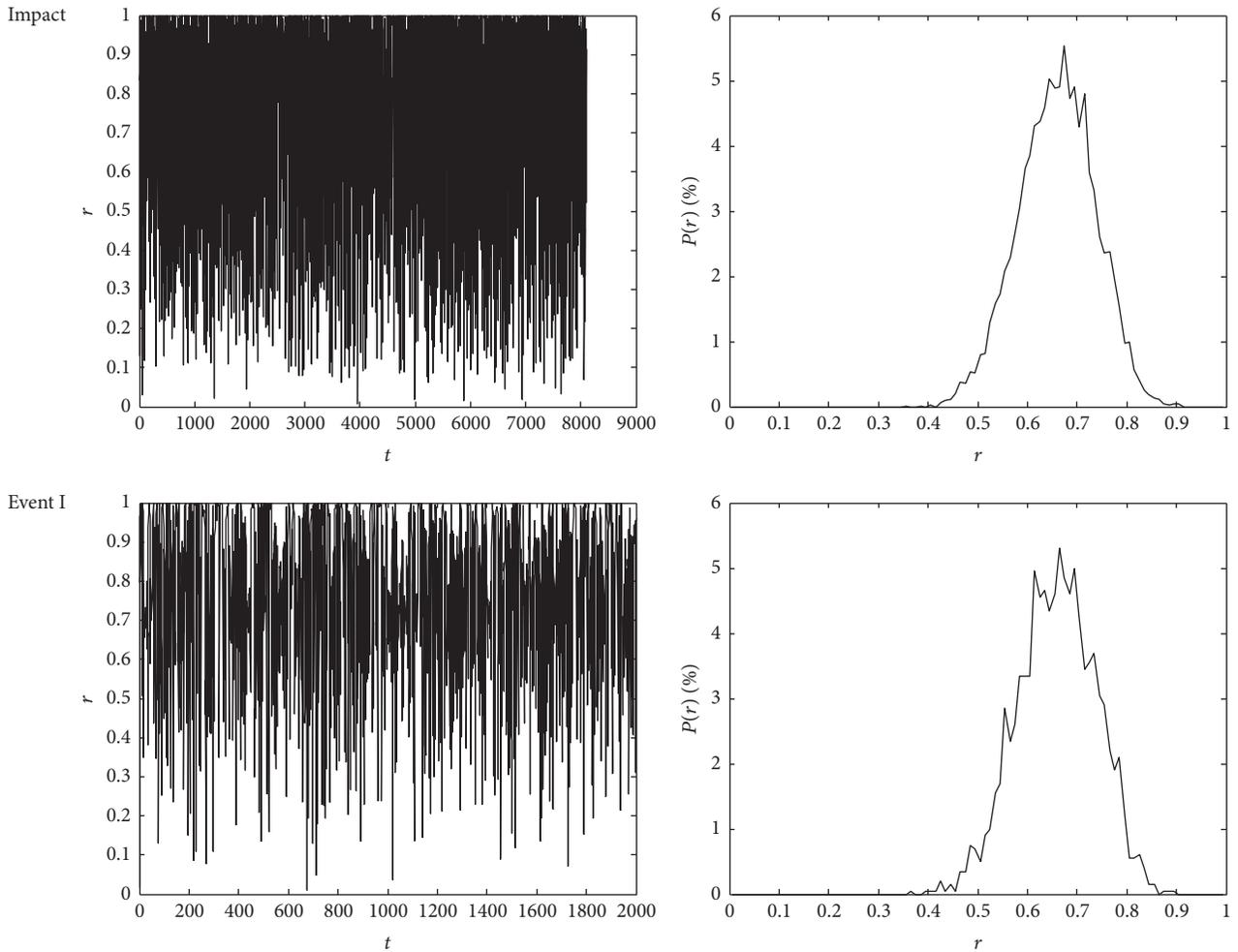


FIGURE 7: Variations of r and the corresponding histogram, from the impact test, event B, and event I.

where n is the reference blade and j are the remaining 21 blades.

In this way, discrepancies among similar signals can be identified in time and space. Since the function is symmetric, only half of it is used in the analysis. With the construction of these maps, it is possible to identify three independent variables: a short time (lag order), a long time (defined in Table 1 as event), and the space (relative position between each blade and blade 1). Based on these three variables, contour plots were produced for each measurement data. Figure 8 shows the contour map of the impact test with respect to blade 1. The horizontal axis is the blade number j in (7), the vertical axis is the lag order (τ), and the contour plot represents the magnitude of C .

As in the correlation factor, there are three dominant regions or clusters (vertical strips) at blades 6, 14, and 22. From the contour plots, it is easily noticed that the correlation amplitude increases linearly with respect to lag. Nevertheless, there are variations at certain blades.

After analyzing the impact test, the cross-correlation functions of the ten airflow measurements were calculated. These calculations produced 10×22 maps where it is possible

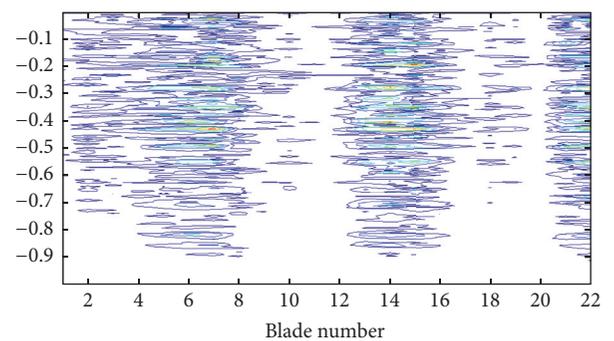


FIGURE 8: Correlation function map for the lag order as a function of blade number, corresponding to the impact test. The correlation is computed with respect to blade 1 ($n = 1$).

to see variations in time and space. As an illustration, the ten maps corresponding to blade 1 ($n = 1$) are presented in Figure 9.

The maps show three different variations, the maximum correlation value, the density of contour plots, and the location of the maxima. This suggests that synchronization

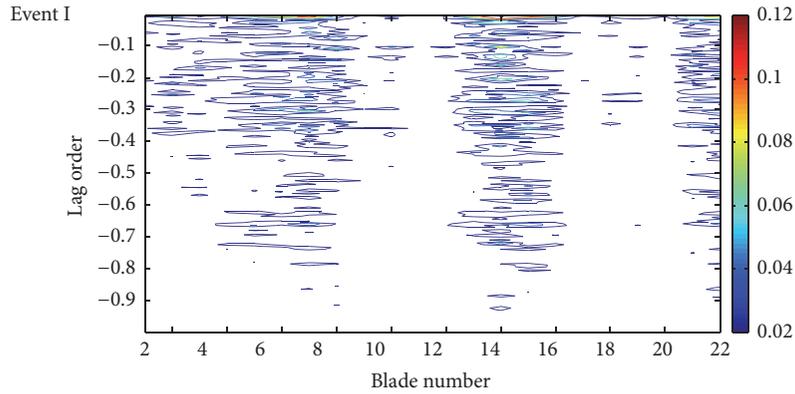


FIGURE 9: Cross-correlation map for blade 1, event I.

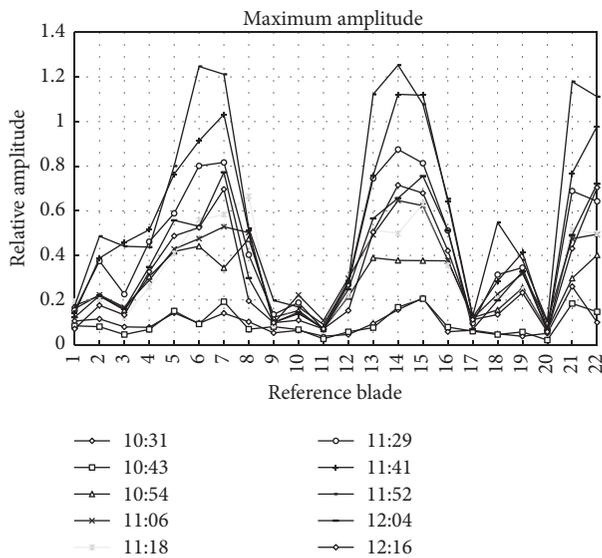


FIGURE 10: Variations of the maximum correlation as a function of the blade location for different events.

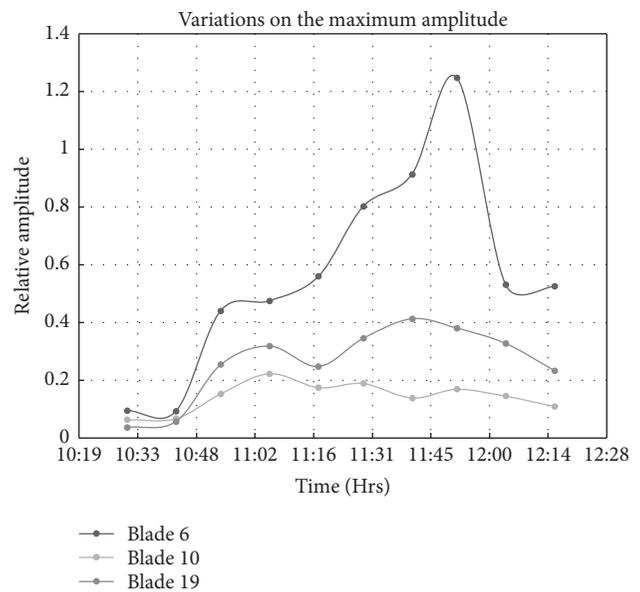


FIGURE 11: Variations of the maximum amplitude as a function of time (all events), blade number 6.

varies in time and space. Determining these variations is not straightforward, and for that reason the four different analyses were conducted. The first consisted of evaluating the variation of the maximum correlation value. Figure 10 shows the blade number where the maximum correlation value occurred, each curve corresponding to an event. Most of the maximum values occur at the same blade location and they evolve following the same pattern, except for blade 10. Figure 11 shows the evolution of the maximum correlation of blades 6, 10, and 19. The maximum has an incremental tendency as the test time increases, whereas the maximum correlation of blade 10 decreases. After comparing these figures, it seems that the internal interaction of the background components starts at blade 10 and propagates to the adjacent elements.

7. Recurrence Plots

Another technique of analysis that can identify synchronization is the recurrence plot. Recurrence is a concept related

to “memory,” if the behavior of a system can be predicted; it means that the system returns to a former state. Poincaré proposed the original concept, and Eckmann introduced the concept of recurrence plots. In general, from Hamilton’s principle, the trajectory along the phase space depends on the velocity and the displacement. The phase space of a simple harmonic has the shape of a helix along the time axis direction. According to Marwan et al. [18], the recurrence plot determines the evolution of the system and finds the number of times that a signal passes through the same point in the phase diagram. If two signals are synchronized, their phase diagrams would be identical. The recurrence plot is a graphical representation of a matrix formed with all the recurrence values. A pure harmonic system shows evenly spaced diagonals. However, for nonharmonic systems, the shorter the diagonals the more chaotic the system. Marwan et al. probed the theory using the Poincaré recurrence theorem. Nevertheless, the theorem is insufficient for predicting the time for a system to return, that is, the recurrence. The

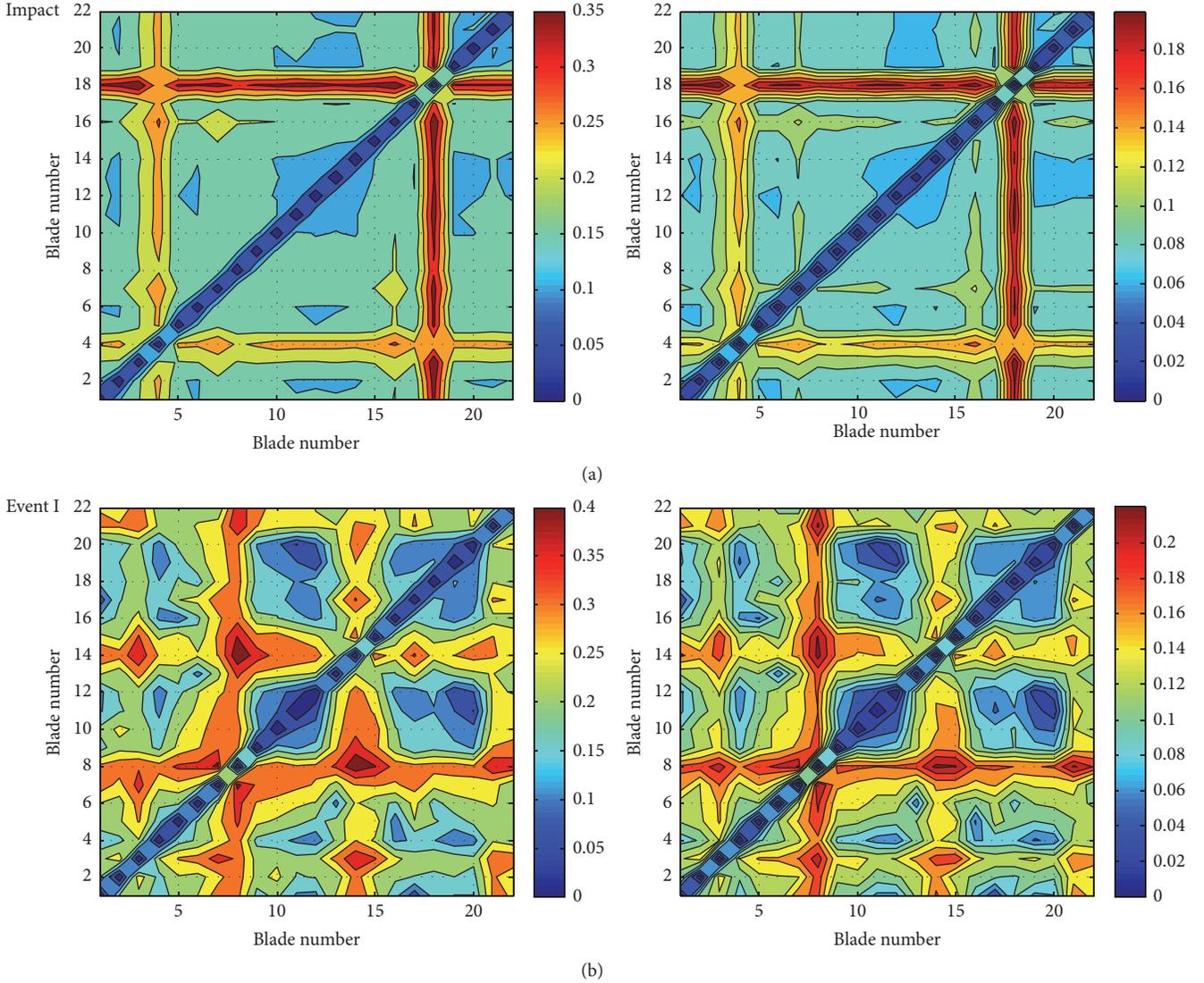


FIGURE 12: (a) Mean value of the reference vector; (b) standard deviation.

determination of the time interval has similar properties to the Markov process, and it is related to the Lyapunov exponents and entropies. What they presented is a way of analyzing the length of the diagonals. One of the challenges of this method is the determination of the appropriate time interval, which is the difference between an orbit and the following orbit. The phase trajectory is represented as an evolving path; thus, the time evolution is the derivative of the trajectory with respect to time. In their work, the authors proposed a method to reconstruct the phase space from discrete data with a time delay factor. In practice, this is feasible only when the system response has a single dominant frequency; otherwise, it is necessary to use other techniques to reconstruct the third coordinate. To calculate the matrix, they recommended the use of the maximum norm. Since the recurrence plot is determined from the difference between a point x_i and the subsequent point x_{i+1} on the phase space, they proposed the use of a tolerance error for computing this difference.

In this work, it was found that the construction of the phase diagram from measurement data is cumbersome. Since the data are the acceleration of each blade, they have to be integrated numerically in order to obtain speed and position.

In order to analyze the synchronization between two signals, the data from the two vectors are analyzed at the same time span, the two phase diagrams are superimposed, and the distance between every point is calculated. In this way, the recurrence plot is constructed using the equation

$$r_{ij} = \sqrt{(p_i - p_j)^2 + (v_i - v_j)^2}. \quad (8)$$

Vectors p_i and v_i are normalized using the maximum value for each case. Ideally, the value of r_{ij} should be zero, but in reality it is impossible. The criterion for determining synchronization is that $r_{ij} < \epsilon$, where ϵ is a tolerance value.

As in previous analyses, the first plot corresponds to the impact test. Then, each event is analyzed. Figure 12 shows the

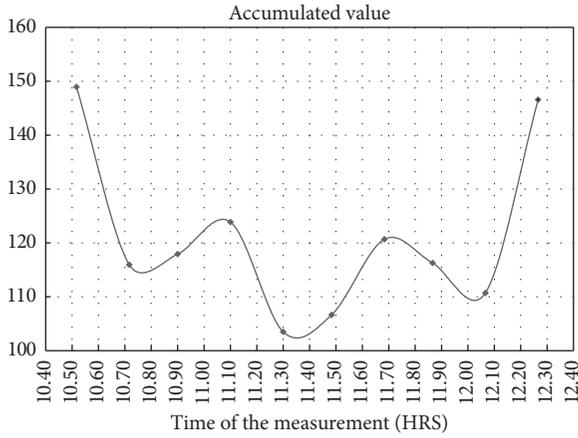


FIGURE 13: Variations on the accumulated value S as a function of time.

recurrence plot of the mean and the standard deviation. The mean value is calculated taking each vector as a reference, starting with blade 1 and finishing with blade 22. In this way, a 22×22 matrix is obtained:

$$R(n, m) = \overline{r_{ij}}, \quad (9)$$

where n is the reference vector and m is the corresponding vector, i is the instantaneous value of vector n , and j is the instantaneous value of vector m .

It is interesting to notice that the structure of the plots is different. With this analysis, the impact test produces a plot that has a different structure than the other data. In the entire airflow test, the dominant values correspond to blades 6, 14, and 21. Data create clusters around these blades, and the cluster pattern appears in every map, except during the impact test. Nevertheless, the pattern shows variation in every event. These variations are associated with the slow dynamic behavior that is a manifestation of the internal interaction between the background components.

In order to determine the slow dynamic behavior, the sum of individual values is calculated as

$$S = \sum_m \sum_n R_{nm}. \quad (10)$$

This number reflects the amount of synchronization at each event. Ideally, it should be zero, but since there are variations in the individual phase diagrams, it reflects the overall dynamic behavior. Figure 13 shows the variations of S as a function of time.

This figure shows a slow cycle with a period of about 36 to 40 min. Therefore, the internal interaction between background components has a frequency of 3.8×10^{-4} Hz. With this analysis, it was possible to estimate the time variation due to the internal interaction. This depends on time and location; the time variations were identified with the recurrence plot and the location with the cross-correlation function.

8. Conclusions

In this paper, the study of the synchronization of a rotor with 24 blades is presented. 22 acceleration vectors were registered simultaneously, and the rotor was subjected to two types of tests: an impact test and an airflow test. Data were recorded at ten different events, separated by long intervals of time.

The measurement and analysis of field data present different challenges; the length of the data is limited by the data acquisition system, and therefore it is impossible to record continuously the acceleration of each blade over a long period of time. The other challenge is the waveform of the internal interaction; in general, the internal interactions have a very low energy content and very low frequency. The accelerometers are unable to identify these waveforms; thus there is a need for other techniques of analysis that are able to identify this behavior.

Synchronization can be easily identified with the correlation coefficient or with the Kuramoto parameter. It was demonstrated that these techniques show synchronization among the foreground components, but they were unable to establish the internal interaction between the background components, neither in time nor in space. In order to overcome this limitation, the data were analyzed with the cross-correlation function and the recurrence plot.

With the cross-correlation function, the data were processed in a way that variations in time and space were represented as contour maps. These contour maps were constructed with the local time variable, the data source (the blade number), and the long waveform (the ten events). With the cross-correlation function, the source of the internal interaction was identified. The criterion for identifying the source was the discrepancy of the behavior of blades 6 and 19 with respect to the rest of the data. These blades showed greater correlations at early events than the rest of the blades that evolved following a similar pattern.

The recurrence plot was the technique that allowed the analysis of the time evolution. This was constructed from the phase diagrams of the blades. Taking the difference of the phase diagrams among the blades it was possible to determine the amount of synchronization. With these differences, the evolution of the mean value during the ten events was evaluated. The evolution in time showed a very slow dynamic pattern with a very low frequency. This frequency is associated with the time variation due to internal interaction among background components.

It has been shown that it is possible to identify the dynamics of internal interaction in complex systems. This interaction has a very low frequency and energy content; therefore, special analysis should be done. The correlation enables identifying the structure of the system and the evolution of synchronization; Kuramoto's parameter estimates the amount of synchronization. The cross-correlation function allows identifying the location of the source, whereas the recurrence plot identifies the time variations. These analysis techniques provide some insight into the coupling source if they are applied together.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

References

- [1] M. Sen, J. Jauregui-Correa, and C. López-Cajun, "Foreground and background components in separable complex systems," *Systems* 4, no. 27, 2016.
- [2] C. Huygens, "Letter to de Sluse. Letter No. 1333 of Feb. 24, 1665, page 241," in *Oeuvres Complète de Christiaan Huygens. Correspondence*, vol. 5, pp. 1664–1665, Société Hollandaise des Sciences, Martinus Nijhoff, La Haye, 1893.
- [3] J. Peña Ramirez, R. H. B. Fey, K. Aihara, and H. Nijmeijer, "An improved model for the classical Huygens' experiment on synchronization of pendulum clocks," *Journal of Sound and Vibration*, vol. 333, no. 26, pp. 7248–7266, 2014.
- [4] K. P. O'Keeffe, P. L. Krapivsky, and S. H. Strogatz, "Synchronization as aggregation: cluster kinetics of pulse-coupled oscillators," *Physical Review Letters*, vol. 115, no. 6, Article ID 064101, 2015.
- [5] H. Ulrichs, A. Mann, and U. Parlitz, "Synchronization and chaotic dynamics of coupled mechanical metronomes," *Chaos*, vol. 19, no. 4, Article ID 043120, 2009.
- [6] À. Aragonès and O. Guasch, "Ranking paths in statistical energy analysis models with non-deterministic loss factors," *Mechanical Systems and Signal Processing*, vol. 52-53, no. 1, pp. 741–753, 2015.
- [7] J. Woodhouse, "An introduction to statistical energy analysis of structural vibration," *Applied Acoustics*, vol. 14, no. 6, pp. 455–469, 1981.
- [8] R. S. Langley and P. Bremner, "A hybrid method for the vibration analysis of complex structural-acoustic systems," *Journal of the Acoustical Society of America*, vol. 105, no. 3, pp. 1657–1671, 1999.
- [9] P. J. Shorter and R. S. Langley, "Vibro-acoustic analysis of complex systems," *Journal of Sound and Vibration*, vol. 288, no. 3, pp. 669–699, 2005.
- [10] C. Llerena-Aguilar, R. Gil-Pita, M. Rosa-Zurera, D. Ayllón, M. Utrilla-Manso, and F. Llerena, "Synchronization based on mixture alignment for sound source separation in wireless acoustic sensor networks," *Signal Processing*, vol. 118, pp. 177–187, 2016.
- [11] A. Ray, A. Roy Chowdhury, and D. Ghosh, "Effect of noise on chaos synchronization in time-delayed systems: numerical and experimental observations," *Physica A. Statistical Mechanics and its Applications*, vol. 392, no. 20, pp. 4837–4849, 2013.
- [12] A. R. Messina and V. Vittal, "Extraction of dynamic patterns from wide-area measurements using empirical orthogonal functions," *IEEE Transactions on Power Systems*, vol. 22, no. 2, pp. 682–692, 2007.
- [13] M. Rosenblum and A. Pikovsky, "Detecting direction of coupling in interacting oscillators," *Physical Review E*, vol. 64, no. 4, Article ID 045202, 2001.
- [14] R. Friedrich, J. Peinke, M. Sahimi, and M. R. Rahimi Tabar, "Approaching complexity by stochastic methods: from biological systems to turbulence," *Physics Reports. A Review Section of Physics Letters*, vol. 506, no. 5, pp. 87–162, 2011.
- [15] F. Ghasemi, J. Peinke, M. Sahimi, and M. R. R. Tabar, "Regeneration of stochastic processes: an inverse method," *European Physical Journal B*, vol. 47, no. 3, pp. 411–415, 2005.
- [16] M. Siefert, A. Kittel, R. Fribdrich, and J. Peinke, "On a quantitative method to analyze dynamical and measurement noise," *Europhysics Letters*, vol. 61, no. 4, pp. 466–472, 2003.
- [17] A. H. Shirazi, G. R. Jafari, J. Davoudi, J. Peinke, M. R. Rahimi Tabar, and M. Sahimi, "Mapping stochastic processes onto complex networks," *Journal of Statistical Mechanics: Theory and Experiment*, vol. 2009, no. 7, Article ID P07046, 2009.
- [18] N. Marwan, M. C. Romano, M. Thiel, and J. Kurths, "Recurrence plots for the analysis of complex systems," *Physics Reports. A Review Section of Physics Letters*, vol. 438, no. 5-6, pp. 237–329, 2007.
- [19] R. V. Donner, M. Small, J. F. Donges et al., "Recurrence-based time series analysis by means of complex network methods," *International Journal of Bifurcation and Chaos in Applied Sciences and Engineering*, vol. 21, no. 4, pp. 1019–1046, 2011.
- [20] R. V. Donner, Y. Zou, J. F. Donges, N. Marwan, and J. Kurths, "Recurrence networks—a novel paradigm for nonlinear time series analysis," *New Journal of Physics*, vol. 12, no. 3, Article ID 033025, 2010.
- [21] Y. Huo, J.-B. Liu, and J. Cao, "Synchronization analysis of coupled calcium oscillators based on two regular coupling schemes," *Neurocomputing*, vol. 165, pp. 126–132, 2015.
- [22] S.-Y. Kim and W. Lim, "Thermodynamic order parameters and statistical-mechanical measures for characterization of the burst and spike synchronizations of bursting neurons," *Physica A. Statistical Mechanics and its applications*, vol. 438, Article ID 16248, pp. 544–559, 2015.
- [23] X. Xu, J. Zhang, and M. Small, "Superfamily phenomena and motifs of networks induced from time series," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 105, no. 50, pp. 19601–19605, 2008.
- [24] R. S. Rosário, P. T. Cardoso, M. A. Muñoz, P. Montoya, and J. G. Miranda, "Motif-synchronization: a new method for analysis of dynamic brain networks with EEG," *Physica A. Statistical Mechanics and its Applications*, vol. 439, pp. 7–19, 2015.
- [25] L. M. Childs and S. H. Strogatz, "Stability diagram for the forced Kuramoto model," *Chaos*, vol. 18, no. 4, Article ID 043128, 043128, 9 pages, 2008.
- [26] A. J. Oberholster and P. S. Heyns, "Online condition monitoring of axial-flow turbomachinery blades using rotor-axial Eulerian laser Doppler vibrometry," *Mechanical Systems and Signal Processing*, vol. 23, no. 5, pp. 1634–1643, 2009.
- [27] J. C. Jáuregui, M. Sen, and C. López-Cajún, "Experimental characterization of synchronous vibrations of blades," in *Inproceedings of the ASME Turbo Expo 2011, paper GT2011-46105*, paper GT2011-46105, Vancouver, Canada, 2011.